

ISSN 2713-3192  
DOI 10.15622/ia.2024.23.3  
<http://ia.spcras.ru>

ТОМ 23 № 3

ИНФОРМАТИКА  
И АВТОМАТИЗАЦИЯ

INFORMATICS  
AND AUTOMATION



СПб ФИЦ РАН

Санкт-Петербург  
2024



# INFORMATICS AND AUTOMATION

Volume 23 № 3, 2024

Scientific and educational journal primarily specialized in computer science, automation, robotics, applied mathematics, interdisciplinary research

Founded in 2002

---

## Founder and Publisher

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)

---

## Editor-in-Chief

**R. M. Yusupov**, Prof., Dr. Sci., Corr. Member of RAS, St. Petersburg, Russia

---

## Editorial Council

<b>A. A. Ashimov</b>	Prof., Dr. Sci., Academician of the National Academy of Sciences of the Republic of Kazakhstan, Almaty, Kazakhstan
<b>I. A. Kalyaev</b>	Prof., Dr. Sci., Academician of RAS, Taganrog, Russia
<b>Yu. A. Merkurjev</b>	Prof., Dr. Sci., Academician of the Latvian Academy of Sciences, Riga, Latvia
<b>A. I. Rudskoi</b>	Prof., Dr. Sci., Academician of RAS, St. Petersburg, Russia
<b>V. Sgurev</b>	Prof., Dr. Sci., Academician of the Bulgarian Academy of Sciences, Sofia, Bulgaria
<b>B. Ya. Sovetov</b>	Prof., Dr. Sci., Academician of RAE, St. Petersburg, Russia
<b>V. A. Soyfer</b>	Prof., Dr. Sci., Academician of RAS, Samara, Russia

## Editorial Board

<b>O. Yu. Gusikhin</b>	Ph. D., Dearborn, USA
<b>V. Delic</b>	Prof., Dr. Sci., Novi Sad, Serbia
<b>A. Dolgui</b>	Prof., Dr. Sci., St. Etienne, France
<b>M. N. Favorskaya</b>	Prof., Dr. Sci., Krasnoyarsk, Russia
<b>M. Zelezny</b>	Assoc. Prof., Ph.D., Plzen, Czech Republic
<b>H. Kaya</b>	Assoc. Prof., Ph.D., Utrecht, Netherlands
<b>A. A. Karpov</b>	Assoc. Prof., Dr. Sci., St. Petersburg, Russia
<b>S. V. Kuleshov</b>	Dr. Sci., St. Petersburg, Russia
<b>A. D. Khomonenko</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>D. A. Ivanov</b>	Prof., Dr. Habil., Berlin, Germany
<b>K. P. Markov</b>	Assoc. Prof., Ph.D., Aizu, Japan
<b>R. V. Meshcheryakov</b>	Prof., Dr. Sci., Moscow, Russia
<b>N. A. Moldovian</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>V. V. Nikulin</b>	Prof., Ph.D., New York, United States
<b>V. Yu. Osipov</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>V. K. Pshikhopov</b>	Prof., Dr. Sci., Taganrog, Russia
<b>A. L. Ronzhin</b>	Prof., Dr. Sci., Deputy Editor-in-Chief, St. Petersburg, Russia
<b>H. Samani</b>	Assoc. Prof., Ph.D., Plymouth, UK
<b>A. V. Smirnov</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>B. V. Sokolov</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>L. V. Utkin</b>	Prof., Dr. Sci., St. Petersburg, Russia
<b>L. B. Sheremetov</b>	Assoc. Prof., Dr. Sci., Mexico, Mexico

---

**Editor:** A.S. Lopotova

**Interpreter:** Ya.N. Berezina

**Art editor:** N.A. Dormidontova

---

## Editorial office address

SPC RAS, 39 litera A , 14-th line V.O., St. Petersburg, 199178, Russia

e-mail: [ia@spcras.ru](mailto:ia@spcras.ru), web: <http://ia.spcras.ru>

## The journal is indexed in Scopus

The journal is published under the scientific-methodological supervision of Department for Nanotechnologies and Information Technologies of the Russian Academy of Sciences

© St. Petersburg Federal Research Center of the Russian Academy of Sciences, 2024

# ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ

Том 23 № 3, 2024

Научный, научно-образовательный журнал с базовой специализацией в области информатики, автоматизации, робототехники, прикладной математики и междисциплинарных исследований.

Журнал основан в 2002 году

---

## Учредитель и издатель

Федеральное государственное бюджетное учреждение науки  
«Санкт-Петербургский Федеральный исследовательский центр Российской академии наук»  
(СПб ФИЦ РАН)

---

## Главный редактор

Р. М. Юсупов, чл.-корр. РАН, д-р техн. наук, проф., Санкт-Петербург, РФ

---

## Редакционный совет

- |                        |  |
|------------------------|--|
| <b>А. А. Ашимов</b>    | академик Национальной академии наук Республики Казахстан, д-р техн. наук, проф., Алматы, Казахстан |
| <b>И. А. Каляев</b>    | академик РАН, д-р техн. наук, проф., Таганрог, РФ  |
| <b>Ю. А. Меркурьев</b> | академик Латвийской академии наук, д-р, проф., Рига, Латвия  |
| <b>А. И. Рудской</b>   | академик РАН, д-р техн. наук, проф., Санкт-Петербург, РФ   |
| <b>В. Сгурев</b>       | академик Болгарской академии наук, д-р техн. наук, проф., София, Болгария                          |
| <b>Б. Я. Советов</b>   | академик РАО, д-р техн. наук, проф., Санкт-Петербург, РФ   |
| <b>В. А. Соيفер</b>    | академик РАН, д-р техн. наук, проф., Самара, РФ  |

---

## Редакционная коллегия

- |                        |   |
|------------------------|---|
| <b>О. Ю. Гусихин</b>   | д-р наук, Диаборн, США  |
| <b>В. Делич</b>        | д-р техн. наук, проф., Нови-Сад, Сербия                             |
| <b>А. Б. Долгий</b>    | д-р наук, проф. Сент-Этьен, Франция                                 |
| <b>М. Железны</b>      | д-р наук, доцент, Пльзень, Чешская республика                       |
| <b>Д. А. Иванов</b>    | д-р экон. наук, проф., Берлин, Германия                             |
| <b>Х. Каия</b>         | д-р наук, доцент, Утрехт, Нидерланды                                |
| <b>А. А. Карпов</b>    | д-р техн. наук, доцент, Санкт-Петербург, РФ                         |
| <b>С. В. Кулешов</b>   | д-р техн. наук, Санкт-Петербург, РФ                                 |
| <b>К. П. Марков</b>    | д-р наук, доцент, Аizu, Япония                                      |
| <b>Р. В. Мещеряков</b> | д-р техн. наук, проф., Москва, РФ                                   |
| <b>Н. А. Молдовян</b>  | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>В. В. Никулин</b>   | д-р наук, проф., Нью-Йорк, США                                      |
| <b>В. Ю. Осипов</b>    | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>В. Х. Пшихопов</b>  | д-р техн. наук, проф., Таганрог, РФ                                 |
| <b>А. Л. Ронжин</b>    | д-р техн. наук, проф., зам. главного редактора, Санкт-Петербург, РФ |
| <b>Х. Самани</b>       | д-р наук, доцент, Плимут, Соединённое Королевство                   |
| <b>А. В. Смирнов</b>   | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>Б. В. Соколов</b>   | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>Л. В. Уткин</b>     | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>М. Н. Фаворская</b> | д-р техн. наук, проф., Красноярск, РФ                               |
| <b>А. Д. Хомоненко</b> | д-р техн. наук, проф., Санкт-Петербург, РФ                          |
| <b>Л. Б. Шереметов</b> | д-р техн. наук, Мехико, Мексика                                     |

---

Выпускающий редактор: А.С. Лопотова

Переводчик: Я.Н. Березина

Художественный редактор: Н.А. Дормидонтова

---

## Адрес редакции

14-я линия В.О., д. 39, лит. А, г. Санкт-Петербург, 199178, Россия

e-mail: ia@spcras.ru, сайт: <http://ia.spcras.ru>

## Журнал индексируется в международной базе данных Scopus

Журнал входит в «Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертации на соискание ученой степени доктора и кандидата наук»

Журнал выпускается при научно-методическом руководстве Отделения нанотехнологий и информационных технологий Российской академии наук

© Федеральное государственное бюджетное учреждение науки

«Санкт-Петербургский Федеральный исследовательский центр Российской академии наук», 2024  
Разрешается воспроизведение в прессе, а также сообщение в эфир или по кабелю опубликованных в составе печатного периодического издания - журнала «ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ» статей по текущим экономическим, политическим, социальным и религиозным вопросам с обязательным указанием имени автора статьи и печатного периодического издания журнала «ИНФОРМАТИКА И АВТОМАТИЗАЦИЯ»

## CONTENTS

### Information Security

V. Voevodin  
A PRIVATE SEMI-MARKOV MODEL AS A TOOL TO REDUCE THE  
COMPLEXITY OF THE TASK OF ASSESSING THE STABILITY OF THE  
FUNCTIONING OF ELEMENTS OF THE INFORMATION  
INFRASTRUCTURE EXPOSED TO THREATS 611

D. Smirnov, O. Evsutin  
METHODOLOGY FOR COLLECTING DATA ON THE ACTIVITY OF  
MALWARE FOR WINDOWS OS BASED ON MITRE ATT&CK 642

P. Borisov, Yu. Kosolapov  
A METHOD TO QUANTITATIVE COMPARE OBFUSCATING  
TRANSFORMATIONS 684

### Digital Information Telecommunication Technologies

A. Ivutin, A. Novikov, M. Pestin, A. Voloshko  
DECENTRALIZED PROTOCOL FOR ORGANIZING SUSTAINABLE  
INTERACTION BETWEEN SUBSCRIBERS IN NETWORKS WITH HIGH  
DYNAMICS OF TOPOLOGY CHANGES 727

V. Yakimov  
RECOVERY OF DISCRETE-TIME SIGNAL BASED ON THE MOVING  
AVERAGE MODEL AND ESTIMATION OF THE SAMPLES CORRELATION  
IN FORWARD AND REVERSE FORECASTING 766

I.K. Ibraheem, A.I. Abdalameer, A.Z. Hatif Naji  
A GENETIC APPROACH-BASED INTRA CODING ALGORITHM FOR  
H.266/VVC 801

B. Mayorov  
MODELS OF COMPOSITE HARMONIC HALF-WAVES AND THE  
RELATIONSHIP OF TIME SAMPLING WITH THE ENTROPY OF TIME  
PARAMETERS OF SIGNALS 831

A. Surchenko, Yu. Nedbailo  
HARDWARE COMPRESSION METHOD FOR ON-CHIP AND  
INTERPROCESSOR NETWORKS WITH WIDE CHANNELS AND  
WORMHOLE FLOW CONTROL POLICY 859

### Robotics, Automation and Control Systems

F. Gundelakh, L. Stankevich  
CLASSIFICATION OF SPATIAL TEMPORAL PATTERNS BASED ON  
NEUROMORPHIC NETWORKS 886

A. Obukhov, D. Dedov, D. Teselkin, A. Volkov, A. Nazarova  
DEVELOPMENT OF A STRESS-FREE ALGORITHM FOR CONTROL OF  
RUNNING PLATFORMS BASED ON NEURAL NETWORK  
TECHNOLOGIES 909

## СОДЕРЖАНИЕ

### **Информационная безопасность**

В.А. Воеводин  
ЧАСТНАЯ ПОЛУМАРКОВСКАЯ МОДЕЛЬ КАК ИНСТРУМЕНТ  
СНИЖЕНИЯ СЛОЖНОСТИ ЗАДАЧИ ОЦЕНИВАНИЯ УСТОЙЧИВОСТИ  
ФУНКЦИОНИРОВАНИЯ ЭЛЕМЕНТОВ ИНФОРМАЦИОННОЙ  
ИНФРАСТРУКТУРЫ, ПОДВЕРЖЕННОЙ ВОЗДЕЙСТВИЮ УГРОЗ 611

Д.В. Смирнов, О.О. Евсютин  
МЕТОДИКА СБОРА ДАННЫХ ОБ АКТИВНОСТИ ВРЕДНОСНОГО  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОД ОС WINDOWS НА БАЗЕ MITRE  
ATT&СК 642

П.Д. Борисов, Ю.В. Косолапов  
СПОСОБ КОЛИЧЕСТВЕННОГО СРАВНЕНИЯ ОБФУСЦИРУЮЩИХ  
ПРЕОБРАЗОВАНИЙ 684

### **Цифровые информационно-телекоммуникационные технологии**

А.Н. Ивутин, А.С. Новиков, М.С. Пестин, А.Г. Волошко  
ДЕЦЕНТРАЛИЗОВАННЫЙ ПРОТОКОЛ ОРГАНИЗАЦИИ  
УСТОЙЧИВОГО ВЗАИМОДЕЙСТВИЯ АБОНЕНТОВ В СЕТЯХ С  
ВЫСОКОЙ ДИНАМИКОЙ ИЗМЕНЕНИЯ ТОПОЛОГИИ 727

В.Н. Якимов  
ВОССТАНОВЛЕНИЕ ДИСКРЕТНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ  
СИГНАЛА НА ОСНОВЕ МОДЕЛИ СКОЛЬЗЯЩЕГО СРЕДНЕГО И  
ОЦЕНКИ КОРРЕЛЯЦИОННОЙ СВЯЗИ ОТСЧЕТОВ ПРИ ПРЯМОМ И  
ОБРАТНОМ ПРОГНОЗИРОВАНИИ 766

И.Х. Ибрагим, А.И. Абдаламир, А.З. Хатиф Наджи  
ОСНОВАННЫЙ НА ГЕНЕТИЧЕСКОМ ПОДХОДЕ АЛГОРИТМ  
ВНУТРИКОДИРОВАНИЯ ДЛЯ N.266/VVC 801

Б.Г. Майоров  
МОДЕЛИ СОСТАВНЫХ ГАРМОНИЧЕСКИХ ПОЛУВОЛН И СВЯЗЬ  
ДИСКРЕТИЗАЦИИ ВРЕМЕНИ С ЭНТРОПИЕЙ ВРЕМЕННЫХ  
ПАРАМЕТРОВ СИГНАЛОВ 831

А.В. Сурченко, Ю.А. Недбайло  
МЕТОДИКА КОМПРЕССИИ ДАННЫХ В НАКРИСТАЛЬНЫХ И  
МЕЖПРОЦЕССОРНЫХ СЕТЯХ С ШИРОКИМИ КАНАЛАМИ И  
ПОЛИТИКОЙ УПРАВЛЕНИЯ ПОТОКОМ WORMHOLE 859

### **Робототехника, автоматизация и системы управления**

Ф.В. Гунделах, Л.А. Станкевич  
КЛАССИФИКАЦИЯ ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ ПАТТЕРНОВ  
НА ОСНОВЕ НЕЙРОМОРФНЫХ СЕТЕЙ 886

А.Д. Обухов, Д.Л. Дедов, Д.В. Теселкин, А.А. Волков, А.О. Назарова  
БЕССТРЕССОВЫЙ АЛГОРИТМ УПРАВЛЕНИЯ БЕГОВЫМИ  
ПЛАТФОРМАМИ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ 909

В.А. ВОЕВОДИН  
**ЧАСТНАЯ ПОЛУМАРКОВСКАЯ МОДЕЛЬ КАК ИНСТРУМЕНТ  
СНИЖЕНИЯ СЛОЖНОСТИ ЗАДАЧИ ОЦЕНИВАНИЯ  
УСТОЙЧИВОСТИ ФУНКЦИОНИРОВАНИЯ ЭЛЕМЕНТОВ  
ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ,  
ПОДВЕРЖЕННОЙ ВОЗДЕЙСТВИЮ УГРОЗ**

*Воеводин В.А. Частная полумарковская модель как инструмент снижения сложности задачи оценивания устойчивости функционирования элементов информационной инфраструктуры, подверженной воздействию угроз.*

**Аннотация.** Для принятия решения по обеспечению безопасности информационной инфраструктуры (ИИ) в целях ее устойчивого функционирования в условиях воздействия угроз требуется инструмент, позволяющий оценить устойчивость функционирования ее отдельных элементов. Применение полумарковской модели для оценивания устойчивости функционирования элементов ИИ, подверженной воздействию угроз, в прямой постановке сопряжено с ростом сложности описания объекта моделирования (параметрического пространства) в степенной прогрессии от числа учитываемых воздействий, что снижает ее практическую значимость. Однако в научной литературе не обнаружено исследований по снижению сложности полумарковской модели. В статье приведен подход к снижению сложности моделирования посредством принятия корректных допущений при формировании исходных данных. Приведены условия, при которых возможно принять ряд допущений, позволяющих значительно снизить сложность моделирования, платой за это является ограничение области применимости модели. Приводится постановка задачи и модифицированный граф переходов. Новизна постановки задачи заключается в учете ограничений на имеющийся ресурс для восстановления функциональности элемента. Для пояснения физической сущности процесса моделирования приводится мысленный эксперимент с моделью. Для решения задачи были использованы: а) экспертные методы для добытия исходных данных; б) математические модели частных полумарковских процессов; в) методы преобразований Лапласа; г) методы планирования эксперимента. Демонстрация последовательности решения задачи сопровождается иллюстративными примерами и графиками. В результате эксперимента были выявлены закономерности исследуемого процесса, существование которых было доказано формально. Исследование показало, что при принятии мотивированных допущений возможно снизить сложность моделирования. Результаты исследования расширяют знания о приложении методов марковских процессов для оценивания устойчивости функционирования элементов ИИ применительно к условиям воздействия угроз.

**Ключевые слова:** полумарковский процесс, сложность оценивания, преобразования Лапласа, функция живучести, устойчивость функционирования, информационная инфраструктура.

**1. Введение.** На текущий момент мы не очень сильны в объективном оценивании устойчивости функционирования элементов критической информационной инфраструктуры (КИИ). Способность оценивать устойчивость функционирования КИИ применительно к воздействию угроз еще слабее. Без умения

своевременно и объективно оценивать (измерять) уровень устойчивости функционирования мы не способны принять обоснованные решения по обеспечению ее безопасности.

В [1 – 4] приведены фундаментальные теоретические основания для приложения общих марковских потоков событий и показана их связь с марковскими аддитивными процессами и процессами марковского восстановления, рассмотрены важные для приложений частные случаи таких потоков. Анализ публикаций [5 – 11] показывает, что вопросам практического применения методов теории полумарковских процессов посвящено большое число работ. Результаты анализа позволяют утверждать, что в основном предлагается подход, основанный на асимптотических оценках устойчивости, что не может являться не спорным. Рассматриваемые моделируемые процессы, без грубых допущений, нельзя принять стационарными, следовательно, усредненные исходные данные могут давать значительную погрешность в оценках устойчивости. Однако за пределами предмета публикаций осталась конкретизация методов теории полумарковских процессов для условий ограничения на ресурс, с помощью которого поддерживается сам полумарковский процесс. При этом не учитывается возможность перехода процесса в невозвратное состояние по исчерпанию ресурса.

Общая постановка задачи оценивания устойчивости функционирования информационной инфраструктуры (ИИ) к воздействию целенаправленных угроз с учетом ограничений на ресурс приведена в [12 – 14]. В этих публикациях было предложено задачу оценивания устойчивости функционирования ИИ привести к частным задачам оценивания устойчивости функционирования ее отдельных элементов. Зная структуру ИИ, можно перейти от результатов оценивания устойчивости отдельных элементов ИИ к оцениванию ее устойчивости функционирования в целом. Принято допущение, что элементы ИИ и воздействующие угрозы рассматриваются без учета их сущности и внутренней структуры. Предложенный автором в [12] алгоритм решения задачи в общей постановке характеризуется значительной дескриптивной сложностью, которая растет в степенной прогрессии в зависимости от числа воздействий угроз. В этих условиях поиск методов снижения сложности алгоритма становится актуальной задачей.

**2. Постановка задачи.** Пусть задана некая ИИ объекта информатизации, содержащая семейство взаимоувязанных элементов, подверженных воздействию угроз. На этапе проектирования ИИ для каждого из элементов были реализованы специальные меры,

обеспечивающие некую их защищенность от воздействия угроз. В составе ИИ обычно предусматривается система восстановления, обладающая соответствующим производственным потенциалом (ресурсом) для восстановления функциональности отдельных элементов. Система восстановления реагирует на поражения отдельных элементов ИИ восстановлением их функциональности, в результате чего восстанавливается функционал ИИ в целом. При этом расходуется определенный ресурс системы восстановления.

Стационарный процесс функционирования элемента в условиях его старения или незначительных воздействий, предусмотренных нормальными условиями эксплуатации, может прерываться аномальными воздействиями окружающей среды, приводящими к его повреждениям и потере функциональности. К таким воздействиям могут относиться стихийные бедствия, целенаправленные воздействия угроз, которые имеют особенности: а) значительные повреждения элемента, которые приводят либо к его уничтожению, либо к существенному увеличению времени восстановления; б) незначительное время между соседними повторными воздействиями угроз; в) совокупность воздействий на относительно небольшом промежутке времени; г) существенное отличие законов распределения времени до повреждения и времени восстановления от аналогичных законов в нормальных условиях эксплуатации. С учетом этих особенностей применение асимптотических оценок для определения вероятности неповрежденного состояния элемента может привести к ошибкам оценивания.

В этих условиях целесообразно использовать функцию устойчивости элемента  $v(t)$ , характеризующую вероятность сохранения функциональности элемента в момент времени  $t$  в условиях воздействий угроз.

Для снижения сложности алгоритма решения задачи принято допущение, суть которого заключается в том, что функция устойчивости элемента на заданном интервале времени реализации угроз  $(0, T]$  представляется в виде произведения двух *независимых* функций с параметрами  $t$ :

$$v(t, u, \Delta T, \lambda, \mu, \mathbf{P}) = K_{\text{ог}}(t, u, \Delta T) \varphi(t, \lambda, \mu, \mathbf{P}), \quad (1)$$

где  $v(t, u, \Delta T, \lambda, \mu, \mathbf{P})$  – функция устойчивости элемента [13];

$t$  – временной параметр оценивания функции устойчивости;

$u$  – показатель безотказности и восстанавливаемости элемента в штатных условиях применения,  $u = \{T_{\text{н}}, T_{\text{в}}\}$ , где  $T_{\text{н}}$  – средняя



наработка на отказ элемента в штатных условиях;  $T_b$  – среднее время восстановления работоспособности элемента в штатных условиях. Значения частных показателей возможно получить на основе обработки соответствующих статистических данных, полученных в штатных условиях применения или из эксплуатационной документации;

$\Delta T$  – период времени воздействия угроз;

$\lambda = \{F, n, T\}$  – характеристика угрозы, где  $F$  – совокупность функций распределения случайных интервалов времени  $\eta_i$  до очередного воздействия  $i$ -й угрозы,  $F = \{F_i(t)\}$ , где  $F_i(t)$  – функция распределения случайного интервала времени  $\eta_i$  до воздействия  $i$ -й угрозы,  $i = 1, 2, \dots, n$ ,  $n$  – количество угроз в серии;  $T$  – момент времени окончания воздействий угроз;

$\mu$  – частный показатель живучести элемента, характеризующий его способность восстанавливать нарушенную, в результате успешного воздействия угроз, функциональность,  $\mu = \{T_b, G\}$ , где  $T_b = \{\tau_{vi}^H, \tau_{vi}^B\}$  – совокупность нижнего и верхнего значений  $i$ -го интервала времени восстановления функциональности;  $G$  – совокупность функций распределения случайных интервалов времени восстановления функциональности элемента после воздействия  $i$ -й угрозы,  $G = \{G_i(t)\}$ ,  $i = 1, \dots, n$ ,  $n$  – число воздействий угроз в серии,  $\tau_{vi}^H$  – оценка нижнего значения интервала восстановления функционала после воздействия  $i$ -й угрозы,  $\tau_{vi}^B$  – оценка верхнего значения интервала восстановления функционала после воздействия  $i$ -й угрозы. Оценка параметров  $\{\tau_{vi}^H, \tau_{vi}^B\}$  осуществляется либо с помощью экспертных методов, либо в результате соответствующих учений, деловых игр или специальных исследований;

$P$  – совокупность частных показателей живучести элемента, характеризующая его защищенность, т.е. способность сохранять требуемый функционал в результате отражения соответствующих угроз. В качестве такого показателя принята совокупность вероятностей поражения элемента при отражении серии угроз,  $P = \{p_i\}$ , где  $p_i$  – вероятность поражения элемента при воздействии  $i$ -й угрозы,  $i = 1, \dots, n$ ,  $n$  – число угроз. Оценивание вероятностей  $p_i$  осуществляется либо с помощью экспертных методов, либо в результате соответствующих учений, деловых игр или статистического моделирования;

$K_{\text{ор}}(t, u, \Delta T) = K_{\Gamma}(u) P(t, \Delta T)$  – нестационарный коэффициент оперативной готовности элемента, определяемый вероятностью его безотказной работы в течение времени реализации серии угроз,  $\Delta T = (0, T]$ , где  $K_{\Gamma}(u)$  – коэффициент готовности элемента, рассчитанный для штатных условий его применения,  $P(t, \Delta T)$  – вероятность безотказной работы элемента в течение периода времени  $\Delta T$ ;

$\varphi(t, \lambda, \mu, \mathbf{P})$  – функция живучести элемента, определяющая вероятность сохранения его функциональности, в условиях воздействия угроз. Для штатных условий ее значение равно единице, так как воздействия угроз отсутствуют.

Для большинства практических случаев значение  $K_{\text{ор}}(u)$  на периоде  $\Delta T$  очень близко к единице, а значение функции живучести на этом же периоде неизмеримо меньше  $\varphi(t, \lambda, \mu, \mathbf{P}) \ll K_{\text{ор}}(u)$ . С учетом этого утверждения формулу (1) можно привести к виду:

$$v(t, \lambda, \mu, u, \mathbf{P}) \approx \varphi(t, \lambda, \mu, \mathbf{P}),$$

т.е. функция устойчивости элемента, для условий воздействия угроз, может быть приближенно заменена функцией живучести.

Для оценивания устойчивости отдельных элементов в [12] была предложена модель общего полумарковского процесса, дескриптивная сложность алгоритма моделирования зависит от числа воздействий угроз и составляет  $m_0 = 3 \cdot 2^n + 1$ . Формула выведена эмпирически. Сложность алгоритма моделирования значительно снижает ее практическую значимость. При этих условиях актуальной становится задача снижения сложности моделирования при допустимой достоверности получаемых результатов. Поэтому основные усилия в рамках настоящего исследования были сосредоточены на поиске пути снижения дескриптивной сложности алгоритма моделирования.

**Цель статьи** – сообщить о предлагаемом подходе к снижению сложности алгоритма моделирования с возможностью контроля достоверности получаемого результата оценивания.

Основная идея метода заключается в том, что если предположить, что воздействия угроз планируются противником в течение равных промежутков времени или через случайные интервалы времени, продолжительность которых определяется последовательностью одних и тех же однотипных операций по подготовке воздействия однотипных угроз, то справедливо принять допущение об одинаковых законах распределения времени до

воздействия угроз  $F_i(t) = F(t)$ , времени восстановления  $G_i(t) = G(t)$  и равных вероятностях поражения  $P_i = P$ . При принятии таких допущений количество необходимой информации для описания алгоритма функционирования элемента значительно сокращается, что в результате позволяет снизить дескриптивную сложность и, как результат, сложность самого моделирования. С учетом принятых допущений общий граф переходов, приведённый в [12], приобретает частный вид, представленный на рисунке 1.

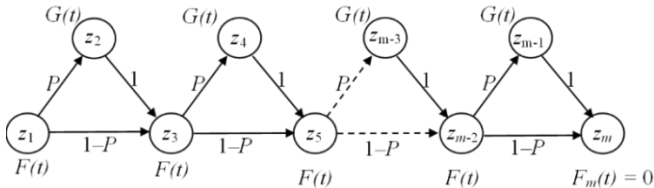


Рис. 1. Граф переходов частного полумарковского процесса

Такой граф переходов (процесс) позиционируется в рамках исследования как частный полумарковский и является конкретизацией общего полумарковского графа, приведенного в [12]. Нечетные индексы  $i = 2j + 1, j = 0, 1, 2, \dots, (n - 1)$ , присваиваются состояниям элемента  $z_i = 1$  («функционален»), четные индексы  $i = 2j, j = 1, 2, \dots, n$ , присваиваются состояниям  $z_i = 0$  («не функционален»). Состояния с индексом  $m = 2n + 1$  позиционируется как поглощающие «функционален по завершению воздействия угроз». Формула для расчета числа возможных состояний графа (дескриптивной сложности), выведенная эмпирически, имеет вид  $m_{\text{ч}} = 2 \cdot n + 1$ .

Процесс завершается восстановлением функциональности элемента при условии, что  $\Pi_{r_n}(t) \leq \Pi_{r_0}(t)$ , где:

а.  $\Pi_{r_0}(t)$  – показатель, характеризующий фактические производственные возможности подсистемы восстановления функционала на момент времени  $t$ :  $\Pi_{r_0}(t) = \{R_0(t), D_0(t)\}$ , где  $R_0(t) = \{r_{0i}(t)\}$  – совокупность типов не возобновляемого ресурса, имеющегося в составе системы восстановления для восстановления элемента на момент времени  $t, i = 1, 2, \dots, m$ ;  $m$  – число типов не возобновляемого ресурса (пример не возобновляемого ресурса – запасные части, расходный материал и т.п.);  $D_0(t) = \{d_{0i}(t)\}$  – совокупность типов возобновляемого ресурса, имеющегося в составе системы восстановления функционала на

момент времени  $t$ ,  $i = 1, 2, \dots, k$ ,  $k$  – число типов возобновляемого ресурса (пример возобновляемого ресурса – человеческий ресурс, измерительные приборы и т.п., которые могут использоваться многократно);

б.  $\Pi_{r_n}(t) = \{R^{Tp}(t), D^{Tp}(t)\}$  – показатель, характеризующий *требуемые* производственные возможности расчета восстановления на момент времени  $t$ , где  $R^{Tp}(t) = \{r_i^{Tp}(t)\}$  – совокупность соответствующих единиц не возобновляемого ресурса каждого типа, требуемого для восстановления функционала элемента после успешного воздействия угроз к моменту времени  $t$ ,  $i = 1, 2, \dots, m$ ,  $m$  – число требуемых типов не возобновляемого ресурса;  $D^{Tp}(t) = \{r_i^{Tp}(t)\}$  – совокупность типов возобновляемого ресурса, требуемого для восстановления функционала элемента к моменту времени  $t$ ,  $i = 1, 2, \dots, k$ ,  $k$  – число требуемых типов возобновляемого ресурса.

Условие перехода процесса в поглощающее состояние «не функционален по истощению ресурса восстановления» характеризуется тем, что требуемый ресурс по любой из составляющих будет больше, чем имеется в наличии на момент времени  $t$ ,  $\Pi_{r_n}(t) > \Pi_{r_0}(t)$ .

Для снижения сложности общего полумарковского процесса, приняты следующие дополнительные допущения:

1. Функция распределения времени пребывания процесса в конечном поглощающем состоянии  $F_m(t) = 0$ . Физически это означает, что воздействия угроз завершились восстановлением функциональности элемента.

2. Одношаговые вероятности перехода  $\pi_{ik}$  элемента из состояния  $z_i$  в состояние  $z_k$  имеют следующие значения:

$\pi_{ik} = 0$  при  $i \geq k$  и  $(k - i) > 2$ , т.к. переход возможен только в состояния, индексы которых превышают индексы предыдущих состояний.

Между индексами  $k$  и  $i$  всегда должно выдерживаться соотношение  $k = i + 1$  при переходе из вершины с нечетным индексом в вершину с четным индексом или  $k = i + 2$  при переходе из вершины с нечетным индексом в вершину с нечетным индексом, в противном случае  $\pi_{ik} = 0$ ;

$$\pi_{ik} = 1 \text{ при } i = 2j, k = 2j + 1;$$

$$\pi_{ik} = P \text{ при } i = 2j - 1, k = 2j;$$

$$\pi_{ik} = 1 - P \text{ при } i = 2j - 1, k = 2j + 1, j = 1, 2, \dots, n.$$

При этом текущее значение функции живучести элемента в момент времени  $t$  будет определяться суммой соответствующих интервально-переходных вероятностей:

$$P_{1i}(t): \varphi(t) = \sum_{\substack{i \in N_{(+)} \\ \Pi_n \leq \Pi_{r_0}}} P_{1i}(t),$$

где  $N_{(+)} = \{2j + 1, j = 0, 1, \dots, n\}$  – множество индексов состояний элемента, при которых  $z_i = 1$  (элемент в состоянии «функционален»).

**3. Мысленный эксперимент.** Мысленный эксперимент приводится для понимания физической сущности процесса моделирования. Пусть у нас имеется идеальная игральная кость с числом граней 6. Грани имеют равные площади, на каждой грани нанесено число  $i$  от 1 до 6. Четные числа ( $i = 2, 4, 6$ ) соответствуют состояниям  $z_i = 0$  («не функционален»). Нечетные числа ( $i = 1, 3, 5$ ) соответствуют состояниям  $z_i = 1$  («функционален»). С помощью этой «аналитической игральной кости» можно произвести моделирование трех  $j = 1, 2, 3$  воздействий с вероятностью поражения  $P = (1 - P) = 0,5$  (грани с идеально равными площадями). Далее осуществляется моделирование для  $t = 0$ , считается, что элемент находился в состоянии  $z_0 = 1$  («функционален»). Предположим, что в течение единичного периода времени  $t$  элемент возвращается в исходное состояние «функционален». Производим первый опыт с бросанием «аналитической игральной кости», пусть он соответствует моменту  $t = 1$ . Если выпало нечетное число, то элемент остался в состоянии «функционален», в противном случае – «не функционален», и т.д. Проводим серию опытов до выполнения условия  $t = 3, (0, 3]$ . Подсчитываем число комбинаций с нечетными числами, нормируем его относительно  $T = 3$ . Можно сразу бросать  $T = 3$  кости, а потом подсчитывать результат. В результате этого мысленного эксперимента можно предположить, что функция устойчивости будет пропорциональна числу событий с выпадением нечетных чисел в рамках этой реализации, что можно принять как гипотезу. Если реализовать несколько таких серий, то результат можно усреднить по числу реализаций. Это тоже можно принять как гипотезу. Также в качестве гипотезы можно принять утверждение, что если площадь соответствующих граней ставить в соответствие со значением вероятности наступления события, то в результате опыта можно получить функцию живучести для различных вероятностей. Например, если потребуется разыграть вероятность наступления события объект

«функционален» с вероятностью  $P = 1$ , то для опыта можно использовать игральную кость в виде тетраэдра с гранями 1, 3, 5.

Понимание сущности мысленного опыта позволяет построить аналитическую модель для расчета значений  $P_{1i}(t)$ , которая представляет собой систему линейных интегральных уравнений. Общий вид уравнений и их вывод приведен в [12]. При моделировании учитывались результаты приложений теоретических результатов марковских процессов для решения конкретных задач из области устойчивости сложных систем, радиотехники, автоматики, теории надежности и массового обслуживания, приведенные в [15 – 34]. Сведения о возможных воздействиях угроз, которые также учитывались при моделировании, приведены в [35].

**4. Решение задачи.** Для решения задачи система уравнений, приведенная в [2], с учетом принятых ограничений, была преобразована в систему линейных интегральных уравнений следующего вида:

$$P_{1,i}(t) = [1 - F(t)]\delta_{1i} + \sum_{\substack{k \in N_{(+)} \\ \Pi_n \leq \Pi_{r_0}}} \pi_{1k} \int_0^t f_{1k}(\tau) P_{ki}(t - \tau) d\tau, \quad (2)$$

где  $i \in N_{(+)}$ ,  $N_{(+)} = \{2j + 1, j = 0, 1, \dots, n\}$  – множество индексов состояний элемента, при которых  $z_i = 1$  (рисунок 2 – это нижние узлы);  $\delta$  – символ Кронекера:  $\delta_{1i} = 1$  при  $i = 1$ ;  $\delta_{1i} = 0$  при  $i > 1$ ;  $k \in N_{(+)}$ ;  $\pi_{1k}$  – вероятность перехода из состояния с индексом 1 в состояние с индексом  $k$ ;  $f_{1k}(\tau)$  – условная плотность распределения времени пребывания элемента в состоянии  $z_k$ ;

$$P_{ki}(t) = [1 - F(t)]\delta_{ki} + \sum_{\substack{r \in N_{(-)} \\ \Pi_n \leq \Pi_{r_0}}} \pi_{kr} \int_0^t f_{kr}(\tau) P_{ri}(t - \tau) d\tau,$$

где  $P_{ki}$  – вероятность перехода процесса в момент времени  $t$  из состояния с индексом  $k$  в состояние с индексом  $i$ , переход осуществляется через верхний узел, рисунок 1;  $\delta_{ki}$  – символ Кронекера,  $\delta_{ki} = 1$  при  $k = i$  и  $\delta_{ki} = 0$  при  $k \neq i$ ;  $r \in N_{(-)}$ ,  $N_{(-)} = \{2j, j = 1, \dots, n\}$  – множество индексов состояний элемента, при которых  $z_r = 0$ , рисунок 1 – это верхние узлы,  $r > k$ ;  $\pi_{kr}$  – вероятность перехода из состояния с индексом  $k$  в состояние с индексом  $r$ ;

$f_{kr}(\tau)$  – условная плотность распределения времени пребывания процесса в состоянии  $z_k$ ;  $P_{ri}$  – вероятность перехода процесса в момент времени  $t$  из состояния с индексом  $r$  в состояние с индексом  $i$ .

Первый член справа в уравнении (2) позволяет учесть вероятность того, что процесс останется в состоянии  $z_1$  до момента времени  $t$ .

Второй член в уравнении (2) отображает вероятность последовательности событий, когда процесс совершает последовательный переход (развивается) из состояния 1 в состояния  $k$  за период времени  $\tau$  и, за оставшееся время  $(t - \tau)$ , из состояния  $k$  в состояния  $r$ . Вероятности частных переходов суммируются по всем промежуточным состояниям  $z_k$ , в которые возможны переходы из состояния  $z_1$ , и интегрируются по возможным временам перехода, которые изменяются от 0 до  $t$ .

При этом в качестве показателя устойчивости элемента может быть использовано значение не только вероятности сохранения элементом своей функциональности, но и вероятность того, что будет успешно отражено заданное число воздействий угроз при имеющемся ресурсе  $\Pi_{r,0}(t)$ . Вторая постановка рассчитана на перспективу исследований.

Таким образом, в соответствии с постановкой задачи, система линейных интегральных уравнений (1) позволяет выразить интервально-переходные вероятности через основные параметры полумарковского процесса, которые предполагается добывать экспертными методами. Однако в виде, представленном формулой (2), модель имеет лишь теоретическое значение и не может быть непосредственно применена для оценивания функции живучести  $\phi(t)$  для произвольного числа  $n$ .

При аналитическом моделировании следует учитывать, что получение формальных выражений для исследуемого процесса в прямой постановке (2) является весьма трудоемкой задачей, сами выражения с трудом поддаются интерпретации лицом, принимающим решение. Если аналитические модели оказываются сложными для реализации, то на практике для построения функции живучести целесообразно применять методы статистического моделирования [36]. Однако решение задачи с помощью аналитического моделирования имеет свою практическую и теоретическую значимость, которая заключается в том, что исследователю представляется возможность получить результаты с помощью и аналитического, и статистического моделирования, сравнить полученные результаты на малых выборках и оценить тем самым расхождение получаемых результатов.

Однако, если удастся автоматизировать процедуру построения линейных интегральных уравнений (2), аналитическая модель может быть применена для практических целей. В настоящее время разработана программа для ЭВМ, позволяющая автоматизировать процесс построения графа переходов, подана заявка на ее регистрацию в Роспатент (зарегистрирована Роспатентом под номером 2023683185).

Некоторое технологическое преимущество для аналитического моделирования дает применение метода преобразований Лапласа [2 – 4]. Чтобы воспользоваться этим преимуществом, система уравнений (2), с учетом принятых допущений, была преобразована к следующему виду:

$$P_{kk}(t) = 1 - F(t), \quad k = 2j + 1, \quad j = 0, 1, 2, \dots, n - 1;$$

индекс  $kk$  учитывает вероятность того, что процесс достигнет состояния с индексом  $k$  и останется в этом состоянии  $z_k$  до момента времени  $t$ :

$$P_{1,(2j+1)} = P \int_0^t f(\tau) P_{2,(2j+1)}(t - \tau) d\tau + (1 - P) \int_0^t f(\tau) P_{3,(2j+1)}(t - \tau) d\tau,$$

где  $j = 1, 2, \dots, n$  при этом  $\Pi_{r_n} \leq \Pi_{r_0}$ .

Индексы  $1, (2j + 1)$  физически отображают траекторию перехода из состояния с индексом 1 в состояние  $(2j + 1)$ , а сама вероятность, что элемент останется в этом состоянии  $(2j + 1)$  до момента времени  $t$ :

$$P_{k,(2j+1)}(t) = \int_0^t g(\tau) P_{(k+1),(2j+1)}(t - \tau) d\tau,$$

где  $k = 2, 4, \dots, 2j; j = 1, 2, \dots, n; \Pi_{r_n} \leq \Pi_{r_0}$ . Индексы  $k, (2j + 1)$  физически отображают траекторию перехода из состояния с индексом  $k$  (из состояния «не функционален» в состояние «функционален») в состояние  $(2j + 1)$ , а вероятность, что элемент останется в этом состоянии  $(2j + 1)$  до момента времени  $t$ , при этом  $k$  – это *четные* индексы;

$$P_{k,(2j+1)}(t) = P \int_0^t f(\tau) P_{(k+1),(2j+1)}(t - \tau) d\tau + (1 - P) \int_0^t f(\tau) P_{(k+2),(2j+1)}(t - \tau) d\tau,$$



где  $k = 3, 5, \dots, (2j - 1); j = 1, 2, \dots, n; \Pi_{r_n} \leq \Pi_{r_0}$ . Индексы  $k, (2j + 1)$  физически отображают траекторию перехода из состояния с индексом  $k$  (остается в состоянии «функционален») в состояние  $(2j + 1)$ , а сама вероятность показывает, что элемент останется в этом состоянии  $(2j + 1)$  до момента времени  $t$ , при этом  $k$  – это *нечетные* индексы;

$$P_{mm}(t) = 1, m = 2n + 1, mm – индекс поглощающего состояния.$$

Полученная система интегральных уравнений позволяет связать посредством преобразования Лапласа [2] интервально-переходные вероятности с основными, в том числе и управляемыми, характеристиками процесса поддержания функциональности элемента в условиях воздействия угроз.

Таким образом, применяя к элементам системы уравнений преобразование Лапласа, получаем следующие отображения:

$$\mathcal{L}\{P_{kk}(t)\} = \tilde{P}_{kk}(s) = s^{-1} [1 - \tilde{f}(s)], \text{ при } k = 2j + 1, j = 0, 1, 2, \dots, n - 1,$$

где  $\mathcal{L}$  – символ прямого интегрального преобразования Лапласа,  $s$  – комплексная переменная изображения при преобразовании Лапласа, “ $\sim$ ” – символ над функцией комплексного переменного  $s$  (изображение), получаемого в результате прямого преобразования Лапласа функции вещественного переменного  $t$  (оригинала):

$$\tilde{P}_{kk} s^{-1} [1 - \tilde{f}(s), k = 2j + 1, j = 0, 1, 2, \dots, n - 1];$$

$$\tilde{P}_{1,(2j+1)}(s) = P \tilde{f}(s) \tilde{P}_{2,(2j+1)}(s) + (1 - P) \tilde{f}(s) \tilde{P}_{3,(2j+1)}(s), \quad (2)$$

где  $j = 1, 2, \dots, n - 1$ ;

$$\tilde{P}_{k,(2j+1)}(s) = \tilde{g}(s) \tilde{P}_{(k+1),(2j+1)}(s),$$

где  $k = 2, 4, \dots, 2j, j = 1, 2, \dots, n$ ;

$$\tilde{P}_{k,(2j+1)}(s) = \tilde{f}(s) [P \tilde{P}_{(k+1),(2j+1)}(s) + (1 - P) \tilde{P}_{(k+2),(2j+1)}(s)],$$

где  $k = 3, 5, \dots, 2j - 1$ ;

$$\tilde{P}_{mm}(s) = s^{-1},$$

где  $m = 2n + 1$ .

Последовательно подставляя выражения  $\tilde{P}_{k,(2j+1)}(s)$ , для  $k > 1$  в (2), определяем:

$$\tilde{P}_{1,(2j+1)}(s) = s^{-1} [1 - \tilde{f}(s)] \tilde{f}^j(s) [1 - P + P\tilde{g}(s)]^j, \quad j = 1, 2, \dots, (n-1);$$

$$\tilde{P}_{1,m}(s) = s^{-1} \tilde{f}^n(s) [1 - P + P\tilde{g}(s)]^n, \quad \text{при } m = 2n + 1.$$

В итоге, используя соотношение  $\tilde{\varphi}(s) = \sum_{j=0}^n \tilde{P}_{1,2j+1}(s)$ , получаем преобразование Лапласа функции живучести следующего вида:

$$\begin{aligned} \tilde{\varphi}(s) = & s^{-1} [1 - \tilde{f}(s)] + s^{-1} [1 - \tilde{f}(s)] \sum_{j=1}^{n-1} \tilde{f}^j(s) [1 - P + P\tilde{g}(s)]^j + \\ & + s^{-1} \tilde{f}^n(s) [1 - P + P\tilde{g}(s)]^n, \end{aligned} \quad (3)$$

где  $n$  – число угроз в серии воздействий.

Для определения обратного преобразования Лапласа выражение (3) было представлено в виде многочлена:

$$\tilde{\varphi}(s) = b_1 + b_1\alpha + b_1\alpha^2 + \dots + b_1\alpha^{n-1} + h\alpha^n,$$

где  $b_1 = s^{-1} [1 - \tilde{f}(s)]$ ,  $\alpha^j$  определяются по формуле:

$$\alpha^j = \tilde{f}^j(s) \sum_{k=0}^j C_j^k (1-P)^{(j-k)} P^k \tilde{g}^k(s), \quad j = 1, 2, \dots, n,$$

где  $C_j^k$  – число сочетаний из  $j$  элементов по  $k$ ,  $h = s^{-1}$ .

Объединив члены, содержащие произведения:

$$s^{-1} \tilde{f}^{j+r}(s) \tilde{g}^j(s), \quad j = 1, 2, \dots, n; \quad r = 0, 1, 2, \dots, n-j;$$

$$s^{-1} \tilde{f}^{j+r+1}(s) \tilde{g}^j(s), \quad j=1, 2, \dots, n; \quad r=0, 1, 2, \dots, n-j-1,$$

получаем:

$$\tilde{\Phi}(s) = s^{-1} - \tilde{V}_1(s) + \sum_{j=1}^{n-1} [\tilde{\Phi}_j(s) - \tilde{V}_{j+1}(s)] - \tilde{\Phi}_n(s), \quad (4)$$

где:

$$\tilde{V}_1(s) = s^{-1} P \sum_{r=0}^{n-1} (1-P)^r f^{r+1}(s); \quad (5)$$

$$\tilde{\Phi}_j(s) = s^{-1} P^j \sum_{r=0}^{n-j} C_{j+r-1}^{j-1} (1-P)^r \tilde{f}^{j+r}(s) \tilde{g}^j(s), \quad j=1, 2, \dots, n; \quad (6)$$

$$\tilde{V}_{j+1}(s) = s^{-1} P^{j+1} \sum_{r=0}^{n-j-1} C_{j+r}^j (1-P)^r \tilde{f}^{j+r+1}(s) \tilde{g}^j(s), \quad j=1, 2, \dots, n-1. \quad (7)$$

Используя таблицы преобразований Лапласа, получаем следующие соответствия между изображениями и оригиналами:

$$s^{-1} \rightarrow 1; \\ s^{-1} \tilde{f}^{j+r}(s) \tilde{g}^j(s) \rightarrow F^{(j+r)}(t) * g^{(j)}(t),$$

где знаком «\*» обозначена операция свертки функций  $F^{(j+r)}(t)$  и  $g^{(j)}(t)$  соответственно,

$F^{(j+r)}(t)$  –  $(j+r)$ -кратная свертка функции  $F(t)$ , определяемая с помощью рекуррентного соотношения:

$$F^{(j+r)}(t) = \int_0^t F^{(j+r-1)}(t-x) f(x) dx, \quad (8)$$

$g^{(j)}(t)$  –  $j$ -кратная свертка функции  $g(t)$ , определяемая с помощью рекуррентного соотношения (между (5) и (9)):

$$g^{(j)}(t) = \int_0^t g^{(j-1)}(t-x) g(x) dx. \quad (9)$$

Применяя обратное преобразование Лапласа к (4)–(7) и используя соотношения (8), (9) получаем:

$$\begin{aligned} \varphi(t) &= 1 - V_1(t) + \sum_{j=1}^{n-1} [\Phi_j(t) - V_{j+1}(t)] + \Phi_n(t), \text{ или} \\ \varphi(t) &= 1 - \sum_{j=1}^n [V_j(t) - \Phi_j(t)], \end{aligned} \quad (10)$$

где:

$$V_1(t) = P \sum_{r=0}^{n-1} (1-P)^r F^{(r+1)}(t); \quad (11)$$

$$V_j(t) = P^j \sum_{r=0}^{n-j} C_{j+r}^{j-1} (1-P)^r F^{(j+r)}(t) * g^{(j-1)}(t), \quad j > 1; \quad (12)$$

$$\Phi_j(t) = P^j \sum_{r=0}^{n-j} C_{j+r-1}^{j-1} (1-P)^r F^{(j+r)}(t) * g^{(j)}(t), \quad j = 1, 2, \dots, n. \quad (13)$$

В итоге общее выражение для функции живучести восстанавливаемого элемента приобретает вид:

$$\varphi(t) = 1 - \sum_{j=1}^n P^j \sum_{r=0}^{n-j} C_{j+r-1}^{j-1} (1-P)^r Q_{jr}(t), \quad (14)$$

где:

$$\begin{aligned} Q_{jr}(t) &= F^{(j+r)}(t) * g^{(j-1)}(t) - F^{(j+r)}(t) * g^{(j)}(t) = \\ &= \int_0^t 1 - G(t-x) y_{(j-1),r}(x) dx; \end{aligned} \quad (15)$$

$$y_{(j-1),r}(x) = f^{(j+r)}(x) * g^{(j-1)}(x), g^{(0)} = 1.$$

Значения  $Q_{jr}(t)$  определяют условную вероятность того, что элемент был поврежден и будет восстанавливаться в момент  $t$  после  $(j+r)$  воздействий и  $(j-1)$  предыдущих восстановлений, при этом выполняется неравенство  $R_0 \geq R(t)$ . При  $P=1$  функция живучести будет иметь вид:

$$\begin{aligned} \varphi(t, P=1, \Pi_{r_n}(t) \leq \Pi_{r_0}(t)) &= \\ &= 1 - \sum_{j=1}^n [F^{(j)}(t) * g^{(j-1)} - F^{(j)}(t) * g^{(j)}(t)], g^{(0)}(t) = 1. \end{aligned} \quad (16)$$

Сами по себе формулы (10) по (16) сложны для интерпретации и восприятия, поэтому наряду с общим выражением (14) для функции живучести, при небольшом числе воздействий  $n$ , предпочтительно использовать другое соотношение, получаемое путем суммирования взвешенных функций  $\varphi_{ki}(t, P=1)$ ,  $k = 0, 1, 2, \dots, n$ , где  $k$  – число поражений (восстановлений) элемента. Так как каждому значению  $k$  соответствует  $m_k$  различных вариантов возможных поражений и восстановлений элемента из  $n$  по  $k$  и вероятность каждого варианта  $P_k = P^k(1-P)^{n-k}$ , то:

$$\varphi(t) = \sum_{k=0}^n P^k (1-P)^{n-k} \sum_{j=1}^{m_k} \varphi_{ki}(t, P=1),$$

где  $m_k = C_n^k$ ;  $\varphi_{ki}(t, P=1)$  – функция живучести при  $P=1$  и  $i$ -м варианте из  $n$  воздействий угроз и  $k$  поражений элемента. Если все  $n$  воздействий не приводят к поражению элемента,  $\varphi_{01}(t, P=1) = 1$ .

Выражения для  $\varphi_{ki}$  определяются путем формирования соответствующих вариантов  $k_i$  при заданном числе  $n$  воздействий и  $k$  поражений. Каждому варианту  $k_i$  будет соответствовать граф переходов (рисунок 2), где  $z_j = 1, z'_j = 0, j = 1, 2, \dots, k, z_{k+1}$  – поглощающее состояние.

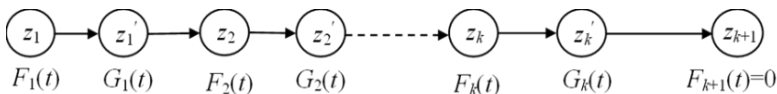


Рис. 2. Граф переходов при  $P = 1$

Вероятности переходов  $\pi_{j,j+1} = 1$ , а функция распределения  $F_j(t)$  времени пребывания процесса в состоянии  $z_j = 1$  будет определяться числом воздействий противника  $n_j$ , которые он должен совершить до  $j$ -го поражения элемента.

В силу соотношения (16) получаем:

$$\varphi_{ki}(t, P = 1) = 1 - \sum_{\substack{j=1, \\ \Pi_n(t) \leq \Pi_{r_0}(t)}}^k \left[ F^{(n_j)}(t) * g^{(j-1)}(t) - F^{(n_j)}(t) * g^{(j)}(t) \right],$$

где  $n_j$  – суммарное число воздействий угроз до  $j$ -го поражения элемента;  $n_j = \sum_{s=1}^j r_s(k_i)$ ;  $r_s(k_i)$  – число воздействий угроз после  $(s-1)$ -го восстановления до  $s$ -го поражения элемента, соответствующее варианту  $k_i$ .

*Новизна результата.* Отличие предлагаемого подхода от приведенных в [2, 4, 12] в том, что: а) для снижения размера параметрического пространства обосновано принимаются дополнительные ограничения, платой за это является некоторое снижение достоверности; б) в возможности учета ограничения на ресурс подсистемы восстановления, платой за это является то, что процесс может перейти в поглощающее состояние по событию «закончился ресурс для восстановления».

**5. Иллюстративный пример.** Пусть дано: число воздействий противника  $n = 3$ , вероятность поражения  $p$  изменяется от 0 до 1 с шагом 0,01. Требуется, используя соотношения (14) и (15), определить общее выражение для функции живучести  $\varphi(t)$ .

При  $n = 3$  возможна реализация  $2^3 = 8$  различных вариантов состояний элемента в процессе воздействия и при заданном числе возможных поражений  $k = 0, 1, 2, 3$  при числе воздействий  $n = 3$ . При этом предполагается, что после каждого поражения элемент восстанавливается до последующего воздействия. Каждый вариант  $k_i$  может быть определен двоичным числом  $b_{ki} = z_1, z_2, z_3$ , где  $k$  – заданное число поражений,  $i$  – номер варианта при заданном  $k$ ,  $z_j$  – состояние элемента при  $j$ -м воздействии противника  $j = 1, 2, 3$ .

Таким образом, при  $k = 0$  элемент подвергался трем воздействиям, но ни разу не был поражен. Число вариантов такого поражения равно единице,  $b_{01} = 000$ , при этом  $\varphi_{01}(t) = 1$ , и вероятность этой ситуации равна  $P_0 = (1 - P)^3$ .

При  $k = 1$  элемент также подвергался трем воздействиям, но поражен всего один раз. Число таких вариантов поражения будет

равно 3, т.к. возможны ситуации, когда объект поражен исключительно после первого воздействия, либо только после второго воздействия или поражен только после третьего воздействия  $b_{11} = 100$ ,  $b_{12} = 010$ ,  $b_{13} = 001$ . Вероятность каждого из этих вариантов поражения равна  $P_1 = P(1 - P)^2$ , при этом для реализации первого варианта требуется число воздействий  $n_1 = 1$ , для второго –  $n_1 = 2$ , для третьего –  $n_1 = 3$ .

При  $k = 2$  число возможных вариантов также равно трем, т.к. элемент может поражаться после первого и второго воздействий, или после первого и третьего, или после второго и третьего, т.е.  $b_{21} = 110$ ,  $b_{22} = 101$ ,  $b_{23} = 011$ . Вероятность каждого из этих вариантов равна  $P_2 = P^2(1 - P)$ , при этом для первого варианта число воздействий  $n_2 = 2$ , для второго варианта  $n_2 = 3$ , для третьего  $n_2 = 3$ .

При  $k = 3$  число вариантов поражения равно единице, т.к. возможно только одна ситуация, когда элемент будет поражен после каждого воздействия, т.е.  $b_{31} = 111$ . Вероятность этого варианта равна  $P_3 = P^3$ . Полученные результаты сведены в таблицу 1.

Таблица 1. Возможные реализации вариантов состояний элемента в результате последовательного воздействия трех угроз

$k$	$i$	$b_{ki}$	$r_1$	$r_2$	$r_3$	$n_1$	$n_2$	$n_3$	$P_k$	$\Phi_{ki}(t)$
0	1	000	0	0	0	0	0	0	$(1-P)^3$	$\Phi_{01}(t)$
1	1	100	1	0	0	1	0	0	$P(1-P)^2$	$\Phi_{11}(t)$
	2	010	2	0	0	2	0	0		$\Phi_{12}(t)$
	3	001	3	0	0	3	0	0		$\Phi_{13}(t)$
2	1	110	1	1	0	1	2	0	$P^2(1-P)$	$\Phi_{21}(t)$
	2	101	1	2	0	1	3	0		$\Phi_{22}(t)$
	3	011	2	1	0	2	3	0		$\Phi_{23}(t)$
3	1	111	1	1	1	1	2	3	$P^3$	$\Phi_{31}(t)$

Используя результаты, приведенные таблице, по формуле (16) определяются  $\Phi_{ki}(t)$  для каждого варианта результатов воздействия:

$$\begin{aligned}
 \varphi_{01}(t) &= 1; \\
 \varphi_{11}(t) &= 1 - F(t) + F(t) * g(t); \\
 \varphi_{12}(t) &= 1 - F^{(2)}(t) + F^{(2)}(t) * g(t); \\
 \varphi_{13}(t) &= 1 - F^{(3)}(t) + F^{(3)}(t) * g(t); \\
 \varphi_{21}(t) &= 1 - F(t) + F(t) * g(t) - F^{(2)}(t) * g(t) + F^{(2)}(t) * g^{(2)}(t); \\
 \varphi_{22}(t) &= 1 - F(t) + F(t) * g(t) - F^{(3)}(t) * g(t) + F^{(3)}(t) * g^{(2)}(t); \\
 \varphi_{23}(t) &= 1 - F^{(2)}(t) + F^{(2)}(t) * g(t) - F^{(3)}(t) * g(t) + F^{(3)}(t) * g^{(2)}(t); \\
 \varphi_{31}(t) &= 1 - F(t) + F(t) * g(t) - F^{(2)}(t) * g(t) + F^{(2)}(t) * g^{(2)}(t) - \\
 &\quad - F^{(3)}(t) * g^{(2)}(t) + F^{(3)}(t) * g^{(3)}(t).
 \end{aligned} \tag{17}$$

По формуле (16) определяется общее соотношение для построения функции живучести:

$$\begin{aligned}
 \varphi(t) &= (1 - P)^3 + P(1 - P)^2 [\varphi_{11}(t) + \varphi_{12}(t) + \varphi_{13}(t)] + \\
 &\quad + P^2(1 - P) [\varphi_{21}(t) + \varphi_{22}(t) + \varphi_{23}(t)] + P^3 \varphi_{31}(t).
 \end{aligned} \tag{18}$$

**6. Результаты эксперимента с моделью.** В соответствии с аналитическими моделями (17) и (18) было произведено тестовое моделирование, результат моделирования представлен на рисунке 3. Также на рисунке 3 приведены результаты имитационного моделирования, которое осуществлялось автором методами Монте-Карло в соответствии с постановкой, приведенной в [35].



Рис. 3. Результаты имитационного и аналитического моделирования



Результаты моделирования, приведенные на рисунке 3, позволяют утверждать, что расхождение результатов аналитического и имитационного моделирования для минимальных значений функции живучести составляют не более  $\pm 2\%$ . Такое совпадение результатов может свидетельствовать о достоверности моделей для определения минимальных значений функции живучести, которые построены на разных исследовательских фундаментах.

На фоне результатов, полученных с помощью опытного наблюдения, была сформулирована гипотеза, справедливость которой была доказана аналитически. Для подтверждения справедливости гипотезы были доказаны следующие лемма и теорема, определяющие монотонность убывания значений функции живучести  $\varphi(t, n)$  на интервале  $(0, T]$  при возрастании числа воздействий  $n$ .

**Лемма 1.** Если интегралы от функций  $y_{j-1}(x) = f^{(j+r)}(x) * g^{(j-1)}(x)$  и  $Y_j(x) = G(t-x)y_{j-1}(x)$ , при  $x \geq 0, j = 1, 2, \dots, n, r = 0, 1, 2, \dots, n-j, g^{(0)}(x) = 1$ , существуют, функция распределения  $G(t-x) < 1$  при  $t < (T+x)$  и выполняется условие  $\Pi_{r_n}(t) \leq \Pi_{r_0}$ , то для любых  $t \in (0, T+x]$  и  $P > 0$  справедливо неравенство:

$$V_j(t) > \Phi_j(t), \tag{19}$$

где  $V_j(t)$  и  $\Phi_j(t)$  определяются по формулам (12) и (13).

Для построения доказательства леммы было проведено моделирование, результаты которого в виде графика приведены на рисунке 4.

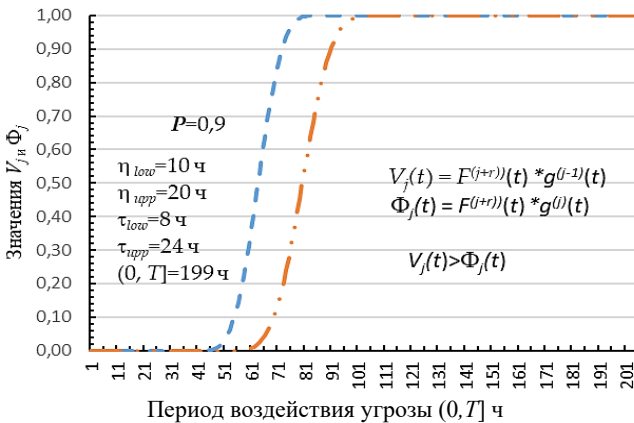


Рис. 4. Иллюстрация к лемме 1

Для доказательства справедливости неравенства (19), учитывая соотношения (10) – (13), что  $F^{(j+r)}(t) * g^{(j-1)}(t) > F^{(j+r)}(t) * g^{(j)}(t)$  при всех  $r = 0, 1, 2, \dots, n-j$  и  $\Pi_{r_0}(t) \leq \Pi_{r_0}$ . Вследствие обратимости операции свертки получаем:

$$F^{(j+r)}(t) * g^{(j)}(t) = G(t) * y_{j-1}(t) = \int_0^t G(t-x) y_{j-1}(x) dx;$$

$$F^{(j+r)}(t) * g^{(j-1)}(t) = \int_0^t y_{j-1}(x) dx = \int_0^t f^{(j+r)}(x) * g^{(j-1)}(x),$$

где  $y_{j-1} = f^{(j+r)}(x) * g^{(j-1)}(x)$  – плотность распределения свертки двух случайных функций, первая из которых представляет собой сумму  $j+r$  случайных величин  $\eta$  – моментов времени воздействия угроз, а вторая – сумму  $j-1$  случайных величин  $\tau$  – моментов времени восстановления функциональности элемента. Так как  $G(t-x) < 1$  при  $t < (T+x)$  и  $y_{j-1}(x) > 0$ , то  $y_{j-1}(x) > G(t-x) y_{j-1}(x)$  при  $t \in (0, T+x]$ . В результате, используя свойство неравенства определенных интегралов, получаем:

$$\int_0^t y_{j-1}(t) dx > \int_0^t G(t-x) y_{j-1}(x) dx \text{ и } F^{(j+r)}(t) * g^{(j-1)}(t) > F^{(j+r)}(t) * g^{(j)}(t).$$

В итоге, сравнивая попарно слагаемые в выражениях с (11) – (13), имеем  $V_j(t) > \Phi_j(t)$  для всех  $j = 1, 2, \dots, n$ ,  $t \in (0, T+x]$  и  $P > 0$  и  $\Pi_{r_0}(t) \leq \Pi_{r_0}$ .

*Примечание.* Очевидно, что неравенство (19) выполняется при  $x=0$ , когда  $G(t) < 1$  и  $t < T$ . В тех случаях, когда функция распределения  $G(t) < 1$  при  $t \in (0, T+x]$ , имеем нестрогое неравенство  $V_j(t) \geq \Phi_j(t)$ . Данная ситуация может иметь место при равномерном законе распределения случайной величины  $\tau$  или при любом другом усеченном законе, когда  $G(t) = 1$  при  $t < T-x$ .

При проведении эксперимента с моделью была сформулирована еще одна гипотеза, суть которой приведена графически на рисунке 5.

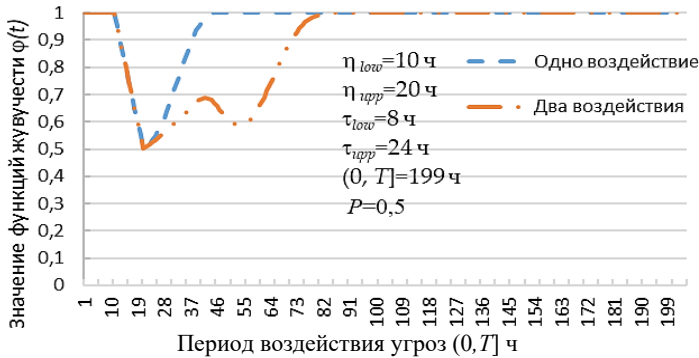


Рис. 5. Сравнение двух функций живучести при различных  $n$

По сути значения функции живучести уменьшается с ростом числа воздействий угроз и не может превосходить значений функции живучести, соответствующей меньшему числу воздействий. Ниже приводится формальное доказательство этой гипотезы.

**Теорема 1.** Для любых  $t \in (0, T]$ ,  $P > 0$  и  $G(t) < 1$  выполняется неравенство  $\varphi(t, n_2) < \varphi(t, n_1)$ , если  $n_2 > n_1$ .

Это означает, что значение функции живучести  $\varphi(t, n_1)$  при одинаковых  $t$  будут выше значений функции живучести  $\varphi(t, n_2)$ . Подставляя значения  $n_1$  и  $n_2$  в формулу (10), получаем:

$$\begin{aligned} \varphi(t, n_2) &= 1 - \left\{ \sum_{j=1}^{n_1} [V_j(t) - \Phi_j(t)] + \sum_{j=n_1+1}^{n_2} [V_j(t) - \Phi_j(t)] \right\} = \\ &= \varphi(t, n_1) - \sum_{j=n_1+1}^{n_2} [V_j(t) - \Phi_j(t)]. \end{aligned}$$

В силу леммы 1, если  $V_j(t) - \Phi_j(t) > 0$  для всех  $j = n_1+1, n_1+2, \dots, n_2$ , то  $\varphi(t, n_2) < \varphi(t, n_1)$  для любых  $t \in (0, T]$ ,  $P > 0$  и  $G(t) < 1$ .

**Следствие 1.1.** Если  $G(t) \leq 1$  при  $t \in (0, T]$  и  $P > 0$ , то наименьшее значение функции живучести монотонно убывает с ростом  $n$ . В силу доказанной ранее теоремы 1  $\varphi(t, n_2) < \varphi(t, n_1)$ , если  $n_2 > n_1$ , поэтому  $\varphi_m(n_2) = \inf_{t \in (0, T]} \varphi(t, n_2) < \varphi_m(n_1) = \inf_{t \in (0, T]} \varphi(t, n_1)$ .

**Теорема 2.** Функция живучести  $\varphi(t)$  на интервале времени воздействия угроз  $(0, T]$  монотонно возрастает при уменьшении среднего времени восстановления функциональности элемента  $m_\tau = M[\tau]$ , т.е. для любых  $t \in (0, T]$  и  $P > 0$  выполняется неравенство

$\varphi(t, m_{\tau_2}) > \varphi(t, m_{\tau_1})$ , если  $G(t, m_{\tau_2}) > G(t, m_{\tau_1})$  и  $M[\tau_1] > M[\tau_2]$ . В силу выражения (15) при выполнении неравенства  $G(x, m_{\tau_2}) > G(x, m_{\tau_1})$  получаем неравенство  $Q_{jr}(t, m_{\tau_2}) < Q_{jr}(t, m_{\tau_1})$  для всех  $j$  и  $r$ . В силу соотношения (14) получаем  $\varphi(t, m_{\tau_2}) > \varphi(t, m_{\tau_1})$ .

График функции живучести (рисунок 6) при верхней оценке времени восстановления  $\tau_{upper} = 24$  ч проходит ниже графика функции живучести для  $\tau_{upper} = 12$  ч, что экспериментально подтверждает формальные выводы теоремы 2.

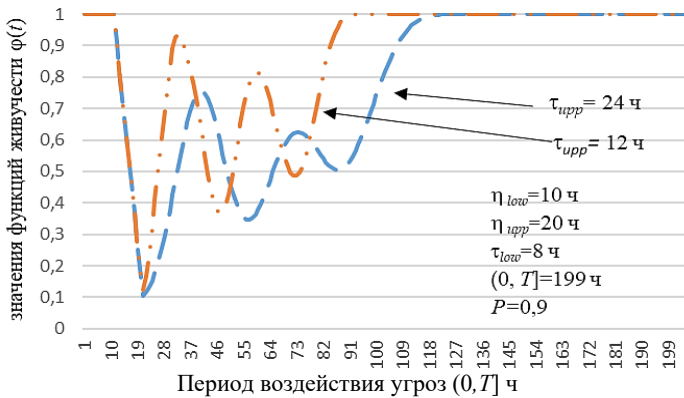


Рис. 6. Иллюстрация к теореме 2

**Следствие 2.1.** Если  $G(t, m_{\tau_2}) > G(t, m_{\tau_1})$  при  $t \in (0, T]$  и  $P > 0$  и  $m_{\tau_2} < m_{\tau_1}$ , то наименьшее значение функции живучести  $\varphi_m(m_{\tau}) = \inf_{t \in (0, T]} \varphi(t, m_{\tau})$  монотонно возрастает с уменьшением  $m_{\tau}$ . Действительно, если уменьшение среднего времени восстановления  $m_{\tau}$  приводит к увеличению значений функции распределения  $G(t, m_{\tau})$  при  $t > 0$ , то в соответствии с теоремой 2 возрастают все значения функции живучести  $\varphi(t)$  на интервале  $(0, T]$ , в том числе и значения  $\varphi_m$ .

**Следствие 2.2.** Пусть  $z^*(t)$  – реализация альтернирующего процесса восстановления элемента.  $S_i$  – площади прямоугольников,  $i = 1, 2, \dots, m$ , образованных ступенчатой кривой  $1 - z^*(t)$  и прямыми  $t = t'_i$  и  $t = t''_i$ , где  $t'_i$  и  $t''_i$  – моменты  $i$ -го поражения и восстановления

элемента соответственно. Альтернирующий процесс проиллюстрирован на рисунке 7.

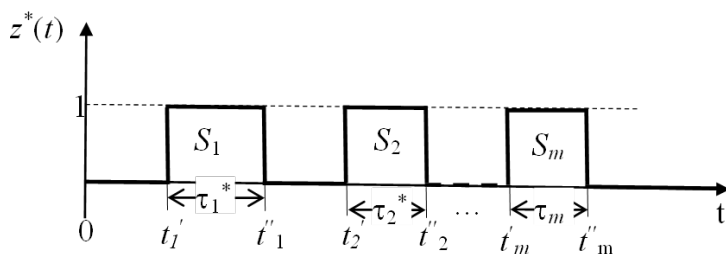


Рис. 7. Иллюстрация к следствию 2.2

Тогда уменьшение любой площади  $S_i$  при каждой реализации  $z^*(t)$  приводит к увеличению наименьшего значения функции живучести  $\varphi(t)$  на интервале  $(0, T]$ .

Действительно, т.к.  $S_i = 1 \cdot \tau_i^*$ , то уменьшение площади  $S_i$  при каждой реализации  $z^*(t)$  однозначно приводит к уменьшению  $\tau_i^*$  времени восстановления  $\tau$  и, соответственно, к уменьшению среднего значения времени восстановления функциональности элемента  $m_\tau$ . В результате, в силу теоремы 2 и следствия 2.1, увеличивается значение  $\varphi_m$ . Из следствия 2.2 вытекает, что минимизация площадей  $S_i$  или значений  $\tau_i^*$  однозначно приводит к увеличению наименьшего значения функции живучести  $\varphi(t)$ .

**7. Заключение.** В результате проведенных исследований можно сделать обоснованное заключение (вывод) о том, что предлагаемый подход для снижения сложности алгоритма оценивания устойчивости функционирования элементов КИИ применительно к воздействию угроз, может быть применен с учетом определенных допущений.

Для оценивания устойчивости возможно применить функцию живучести  $\varphi(t)$  при допущении, что вероятность выхода из строя элемента на периоде  $(0, T]$  ничтожно мала по сравнению с вероятностью потери элементом функциональности в результате воздействия угроз. При моделировании ее можно не учитывать, при этом допускаемая ошибка будет на уровне методического «шума» при экспертном определении исходных данных.

При сравнении итогов аналитического и имитационного моделирования (рисунок 4) следует, что получаемые результаты близки по значению, а расхождение находится на уровне случайной ошибки имитационного моделирования и методического «шума» экспертного определения исходных данных. Эти же результаты

наблюдения (рисунок 4) позволяют косвенно судить о достоверности имитационной и аналитической моделей на основании того, что они сходятся по минимальным значениям на уровне методической ошибки не более чем  $\pm 2$  %. При этом следует учесть, что модели построены с применением принципиально разных подходов.

### Литература

1. Гнеденко Б.В., Беляев Ю.К., Соловьев А.Д. Математические методы в теории надежности. М.: Наука. 1965. 524 с.
2. Тихонов В.И., Миронов М.А. Марковские процессы. М.: «Сов. радио». 1977. 488 с.
3. Королюк В.С. Турбин А.Ф. Полумарковские процессы и их приложения. Киев: «Наукова Думка». 1976. 184 с.
4. Хохлачев Е.Н. Организация и технологии выработки решений при управлении системой и войсками связи. Часть 2. Выработка решений при восстановлении сетей связи. М.: ВА РВСН, 2009. 241 с.
5. Песчанский А.И. Полумарковские модели профилактики ненадежной одноканальной системы обслуживания с потерями. ООО "Научно-издательский центр Инфра-М. 2022. 267 с. DOI: 10.12737/1870597.
6. Привалов А.А., Милашевский А.В. Подход к моделированию узла связи специального назначения, уязвимого к факторам деструктивного воздействия // Инновационная железная дорога. Новейшие и перспективные системы обеспечения движения поездов. Проблемы и решения: Сборник статей Международной научно-теоретической конференции. 2021. С. 308–315.
7. Yu S.-Z. Hidden Semi-Markov Models: Theory, Algorithms and Applications. Morgan Kaufmann, 2015. 208 p.
8. Barbu V.S., Limnios N. Semi-Markov Chains and Hidden Semi-Markov Models toward Applications: their use in Reliability and DNA Analysis. Springer Science and Business Media. 2009. 226 p.
9. Van der Hoek J., Elliott R.J. Introduction to Hidden Semi-Markov Models. Cambridge University Press. 2018. vol. 445. 185 p.
10. Grabski F. Semi-Markov Processes: Applications in System Reliability and Maintenance. Elsevier. 2014. 255 p.
11. Obzherin Yu.E., Boyko E.G. Semi-Markov Models: Control of Restorable Systems with Latent Failures. Academic Press, 2015. 212 p.
12. Воеводин В.А. Модель оценки функциональной устойчивости элементов информационной инфраструктуры для условий воздействия множества компьютерных атак // Информатика и автоматизация. 2023. Т. 22. № 3. С. 691–715. DOI: 10.15622/ia.22.3.8.
13. Воеводин В.А. Генезис понятия структурной устойчивости информационной инфраструктуры автоматизированной системы управления производственными процессами к воздействию целенаправленных угроз информационной безопасности // Вестник Воронежского института ФСИН России. 2023. № 2. С. 30–41.
14. Воеводин В.А., Виноградов И.В., Волков Д.И. Об оценке устойчивости функционирования объекта информатизации в условиях компьютерных атак при экспоненциальном законе распределения времени до воздействия противника и восстановления работоспособности // Вестник Дагестанского государственного технического университета. Технические науки. 2022. Т. 49. № 3. С. 39–51. DOI: 10.21822/2073-6185-2022-49-3-39-51.

15. Шубинский И.Б. Структурная надежность информационных систем. Методы анализа // Ульяновск: Печатный двор, 2012. 216 с.
16. Шубинский И.Б. Функциональная надежность информационных систем. Методы анализа // М.: «Журнал Надежность». 2012. 296 с.
17. Черкесов Г.Н., Недосекин А.О., Виноградов В.В. Анализ функциональной живучести структурно-сложных технических систем // Надежность. 2018. Т. 18. № 2. С. 17–24. DOI: 10.21683/1729-2646-2018-18-2-17-24.
18. Черкесов Г.Н., Недосекин А.О. Описание подхода к оценке живучести сложных структур при многоразовых воздействиях высокой точности // Надежность. 2016. Т. 16. № 2(57). С. 3–15.
19. Антонов С.Г., Анциферов И.И., Климов С.М. Методика инструментально-расчетной оценки устойчивости объектов критической информационной инфраструктуры при информационно-технических воздействиях // Надежность. 2020. Т. 20. № 4. С. 35–41. DOI: 10.21683/1729-2646-2020-20-4-35-41.
20. Захарченко Р.И., Королев Р.И. Методика оценки устойчивости функционирования объектов критической информационной инфраструктуры, функционирующей в киберпространстве // Научные технологии в космических исследованиях Земли. 2018. Т. 10. № 2. С. 52–61.
21. Privalov A., Kotenko I., Saenko I., Evglevskaia N., Titov D. Evaluating the functioning quality of data transmission networks in the context of cyberattacks // Energies. 2021. vol. 14(16). no. 4755. DOI: 10.3390/en14164755.
22. Краснов А.Е., Феоктистова Н.А. Оценка устойчивости критических информационных инфраструктур к угрозам информационной безопасности // Безопасность информационных технологий. 2021. Т. 28. № 1. С. 106–120. DOI: 10.26583/bit.2021.1.09.
23. Oszczypala M., Ziolkowski J., Malachowski J. Analysis of Light Utility Vehicle Readiness in Military Transportation Systems Using Markov and Semi-Markov Processes. Energies. 2022. vol. 15(14). no. 5062. DOI: 10.3390/en15145062.
24. Dhulipala S., Burton H., Baroud H. A Markov Framework for Generalized Post-Event Systems Recovery Modeling: From Single to Multihazards. Structural Safety. 2021. vol. 91. no. 102091. DOI: 10.1016/j.strusafe.2021.102091.
25. Fraccascia L., Giannoccaro I., Albino V. Resilience of Complex Systems: State of the Art and Directions for Future Research. Complexity. 2018. vol. 2018. DOI: 10.1155/2018/3421529.
26. Hu L., Liu X., Zhou K. A Semi-Markov Process Model for Performance Evaluation of DSRC Vehicular Safety Communication. Mathematical Problems in Engineering. 2022. vol. 2022. DOI: 10.1155/2022/7548608.
27. Ouyang M. Review on modeling and simulation of interdependent critical infrastructure systems. Reliability Engineering and System Safety. 2014. vol. 121. pp. 43–60.
28. Helfgott A. Operationalising systemic resilience. European Journal of Operational Research. 2018. vol. 268. no. 3. pp. 852–864.
29. Cere G., Rezgui Y., Zhao W. Critical review of existing built environment resilience frameworks: directions for future research. International Journal of Disaster Risk Reduction. 2017. vol. 25. pp. 173–189.
30. Hosseini S., Barker K., Ramirez-Marquez J.E. A review of definitions and measures of system resilience. Reliability Engineering and System Safety. 2016. vol. 145. pp. 47–61.
31. Righi A.W., Saurin T.A., Wachs P. A systematic literature review of resilience engineering: research areas and a research agenda proposal. Reliability Engineering and System Safety. 2015. vol. 141. pp. 142–152.

32. Li X., Xiao R. Analyzing network topological characteristics of eco-industrial parks from the perspective of resilience: A case study. *Ecological Indicators*. 2017. vol. 74. pp. 403–413.
33. Liu Q., Xing L., Zhou C. Probabilistic modeling and analysis of sequential cyber-attacks. *Engineering Reports*. 2019. vol. 1. no. 4. DOI: 1. 10.1002/eng2.12065.
34. Kordnoori S., Mostafaei H., Kordnoori S., Ostadrahimi M. Testing the Semi Markov Model Using Monte Carlo Simulation Method for Predicting the Network Traffic. *Pakistan Journal of Statistics and Operation Research*. 2020. pp. 713–720. DOI: 10.18187/pjsor.v16i4.3394.
35. ФСТЭК РФ. Методический документ "Методика оценки угроз безопасности информации" (утв. Федеральной службой по техническому и экспортному контролю 5 февраля 2021 г.). URL: <https://fstec.ru/protivodejstvie-korrupsii/metodicheskie-materialy/metodicheskij-dokument> (дата обращения: 21.09.2023).
36. Voevodin V.A. Monte Carlo method for predicting the stability of the functioning of the informatization object in the conditions of massive computer attacks. *Journal of Physics: Conference Series*. 2021. vol. 2099. no. 1. DOI: 10.1088/1742-6596/2099/1/012070.

**Воеводин Владислав Александрович** — канд. техн. наук, доцент кафедры «информационная безопасность», Национальный исследовательский университет «МИЭТ». Область научных интересов: модели, методы и средства обеспечения аудита состояния объекта, находящегося под воздействием угроз нарушения его информационной безопасности, модели и методы оценки защищенности информации, в том числе в интересах страхования информационных рисков, методы оценки эффективности систем (комплексов) мер обеспечения информационной безопасности. Число научных публикаций — 60. [vva541@mail.ru](mailto:vva541@mail.ru); Солнечная Аллея, 901, 124575, Москва-Зеленоград, Россия; р.т.: +7(985)787-1344.

**Поддержка исследований.** При поддержке Фонда Потанина.



V. VOEVODIN

**A PRIVATE SEMI-MARKOV MODEL AS A TOOL TO REDUCE THE COMPLEXITY OF THE TASK OF ASSESSING THE STABILITY OF THE FUNCTIONING OF ELEMENTS OF THE INFORMATION INFRASTRUCTURE EXPOSED TO THREATS**

*Voevodin V. A Private Semi-Markov Model as a Tool to Reduce the Complexity of the Task of Assessing the Stability of the Functioning of Elements of the Information Infrastructure Exposed to Threats.*

**Abstract.** Decision-making on information infrastructure (II) security for its sustainable functioning in the face of threats requires a tool to assess the sustainability of its individual elements. The application of the semi-Markov model to assess the stability of the functioning of elements of II exposed to threats in a direct setting is associated with the increasing complexity of the description of the object of delineation (parametric space) in step progression from the number of the considered impacts, which reduces its practical significance. However, no studies have been found in the scientific literature to reduce the complexity of the semi-Markov model. The article presents an approach to reducing the complexity of modeling by adopting correct assumptions when forming the initial data. Given the conditions under which it is possible to take a series. It was a cost to limit the applicability of the model by significantly reducing the complexity of the modelling. The problem statement and the modified transition graph are given. The novelty of the problem statement is to take into account the limitations on the available resources for the restoration of functionality of the element. To explain the physical essence of the modeling process, a thought experiment with a model is introduced. To solve the problem, the following methods were used: a) expert methods for extraction of initial data; b) mathematical models of private semi-Markov processes; c) methods of transformation of Laplace; r) methods of planning of experiment. Illustrative examples and graphs accompany the task sequence demonstration. As a result of the experiment, the regularities of the studied process, the existence of which was proved formally. As a result of the experiment, the regularities of the studied process were revealed, the existence of which was officially proved. The results of the study broaden knowledge about the application of methods of Markov processes to assess the stability of the functioning of AI elements in relation to the conditions of the impact of threats.

**Keywords:** Semimarkov process, complexity of evaluation, Laplace transformations, survivability function, stability of functioning, information infrastructure.

**References**

1. Gnedenko B.V., Belyaev Yu.K., Solov'ev A.D. *Matematicheskie metody v teorii nadezhnosti* [Mathematical methods in reliability theory]. Moscow: Nauka. 1965. 524 p. (In Russ.).
2. Tihonov V.I., Mironov M.A. *Markovskie process* [Markov processes]. Moscow: «Sov. radio». 1977. 488 p. (In Russ.).
3. Korolyuk V.S., Turbin A.F., *Polumarkovskie processy` i ix prilozheniya*. [Semi-Markov processes and their applications]. Kiev: «Naukova Dumka». 1976. 185 p. (In Russ.).
4. Hohlachev E.N. *Organizacija i tehnologii vyrabotki reshenij pri upravlenii sistemoj i vojskami svjazi. Chast' 2. Vyrabotka reshenij pri vosstanovlenii setej svjazi* [The organization and technologies of decision-making in the management of the communication system and troops. Part 2. Development of solutions for the

- restoration of communication networks]. Moscow: VA RVSN. 2009. 241 p. (In Russ.).
5. Peschanskij A.I. Polumarkovskie modeli profilaktiki nenadezhnoj odnokanalnoj sistemy obsluzhivaniya s poterjami [Semi-Markov models of prevention of unreliable single-channel lossy service system]. OOO "Nauchno-izdatel'skij centr Infra-M". 2022. 267 p. DOI 10.12737/1870597. (In Russ.).
  6. Privalov A.A., Milashevskij A.V. [An approach to modeling a special-purpose communication node vulnerable to destructive factors] Innovacionnaja zhelezjnaja doroga. Novejšie i perspektivnye sistemy obespečenija dvizhenija poezdov. problemy i reshenija: Sbornik statej Mezhduнародnoj nauchno-teoreticheskoj konferencii [An innovative railway. The latest and promising train traffic support systems. Problems and solutions: Collection of articles of the International Scientific and Theoretical Conference]. 2021. pp. 308–315. (In Russ.).
  7. Yu S.-Z. Hidden Semi-Markov Models: Theory, Algorithms and Applications. Morgan Kaufmann, 2015. 208 p.
  8. Barbu V.S., Limnios N. Semi-Markov Chains and Hidden Semi-Markov Models toward Applications: their use in Reliability and DNA Analysis. Springer Science and Business Media. 2009. 226 p.
  9. Van der Hoek J., Elliott R.J. Introduction to Hidden Semi-Markov Models. Cambridge University Press. 2018. vol. 445. 185 p.
  10. Grabski F. Semi-Markov Processes: Applications in System Reliability and Maintenance. Elsevier. 2014. 255 p.
  11. Obzherin Yu.E., Boyko E.G. Semi-Markov Models: Control of Restorable Systems with Latent Failures. Academic Press, 2015. 212 p.
  12. Voevodin V.A. [A Model for Assessing the Functional Stability of Information Infrastructure Elements for Conditions of Exposure to Multiple Computer Attacks]. Informatika i avtomatizacija – Informatics and Automation. 2023. vol. 22. no. 3. pp. 691–715. DOI: 10.15622/ia.22.3.8. (In Russ.).
  13. Voevodin V.A. [The genesis of the concept of structural stability of the information infrastructure of an automated production process management system to the impact of targeted threats to information security]. Vestnik Voronezhskogo instituta FSIN Rossii – Bulletin of the Voronezh Institute of the Federal Penitentiary Service of Russia. 2023. no. 2. pp. 30–41.
  14. Voevodin V.A., Vinogradov I.V., Volkov D.I. [On the assessment of the stability of the functioning of an informatization object in the conditions of computer attacks with an exponential law of time distribution before the enemy's impact and restoration of operability]. Vestnik Dagestanskogo gosudarstvennogo texničeskogo universiteta. Texničeskije nauki – Bulletin of the Dagestan State Technical University. Technical sciences. 2022. vol. 49. no. 3. pp. 39–51. DOI: 10.21822/2073-6185-2022-49-3-39-51. (In Russ.).
  15. Shubinskij I.B. Strukturnaya nadežnost' informacionny'x sistem. Metody analiza [Structural reliability of information systems. Methods of analysis]. Moscow: Zhurnal Nadežnost', 2012. 216 p. (In Russ.).
  16. Shubinskij I.B. Funkcional'naya nadežnost' informacionnyh sistem. Metody analiza [Functional reliability of information systems. Methods of analysis]. Moscow: Zhurnal Nadežnost', 2012. 296 p. (In Russ.).
  17. Cherkesov G.N., Nedosekin A.O., Vinogradov V.V. [Analysis of functional survivability of structurally complex technical systems]. Nadežnost' – Reliability. 2018. vol. 18. no. 2. pp. 17–24. DOI: 10.21683/1729-2646-2018-18-2-17-24. (In Russ.).

18. Cherkesov G.N., Nedosekin A.O. [Description of an approach to assessing the survivability of complex structures under reusable high-precision impacts]. *Nadezhnost' – Reliability*. 2016. vol. 16. no. 2(57). pp. 3–15. (In Russ.).
19. Antonov S.G., Anciferov I.I., Klimov S.M. [The methodology of instrumental and computational assessment of the stability of critical information infrastructure facilities under information technology impacts]. *Nadezhnost' – Reliability*. 2020. vol. 20. no. 4. pp. 35–41. DOI: 10.21683/1729-2646-2020-4-35-41. (In Russ.).
20. Zaharchenko R.I., Korolev I.D. [Methodology for assessing the stability of the functioning of critical information infrastructure facilities operating in cyberspace]. *Naukoemkie tekhnologii v kosmicheskikh issledovaniyah Zemli – High-tech technologies in space exploration of the Earth*. 2018. vol. 10. no. 2. pp. 52–61. (In Russ.).
21. Privalov A., Kotenko I., Saenko I., Evglevskaya N., Titov D. Evaluating the functioning quality of data transmission networks in the context of cyberattacks. *Energies*. 2021. vol. 14(16). no. 4755. DOI: 10.3390/en14164755.
22. Krasnov A.E., Feoktistova N.A. [Assessing the resilience of critical information infrastructures to information security threats]. *Bezopasnost' informacionnyh tekhnologii – Information technology security*. 2021. vol. 28. no. 1. pp. 106–120. DOI: 10.26583/bit.2021.1.09. (In Russ.).
23. Oszczypala M., Ziolkowski J., Malachowski J. Analysis of Light Utility Vehicle Readiness in Military Transportation Systems Using Markov and Semi-Markov Processes. *Energies*. 2022. vol. 15(14). no. 5062. DOI: 10.3390/en15145062.
24. Dhulipala S., Burton H., Baroud H. A Markov Framework for Generalized Post-Event Systems Recovery Modeling: From Single to Multihazards. *Structural Safety*. 2021. vol. 91. no. 102091. DOI: 10.1016/j.strusafe.2021.102091.
25. Fraccascia L., Giannoccaro I., Albino V. Resilience of Complex Systems: State of the Art and Directions for Future Research. *Complexity*. 2018. vol. 2018. DOI: 10.1155/2018/3421529.
26. Hu L., Liu X., Zhou K. A Semi-Markov Process Model for Performance Evaluation of DSRC Vehicular Safety Communication. *Mathematical Problems in Engineering*. 2022. vol. 2022. DOI: 10.1155/2022/7548608.
27. Ouyang M. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety*. 2014. vol. 121. pp. 43–60.
28. Helfgott A. Operationalising systemic resilience. *European Journal of Operational Research*. 2018. vol. 268. no. 3. pp. 852–864.
29. Cere G., Rezguy Y., Zhao W. Critical review of existing built environment resilience frameworks: directions for future research. *International Journal of Disaster Risk Reduction*. 2017. vol. 25. pp. 173–189.
30. Hosseini S., Barker K., Ramirez-Marquez J.E. A review of definitions and measures of system resilience. *Reliability Engineering and System Safety*. 2016. vol. 145. pp. 47–61.
31. Righi A.W., Saurin T.A., Wachs P. A systematic literature review of resilience engineering: research areas and a research agenda proposal. *Reliability Engineering and System Safety*. 2015. vol. 141. pp. 142–152.
32. Li X., Xiao R. Analyzing network topological characteristics of eco-industrial parks from the perspective of resilience: A case study. *Ecological Indicators*. 2017. vol. 74. pp. 403–413.
33. Liu Q., Xing L., Zhou C. Probabilistic modeling and analysis of sequential cyber-attacks. *Engineering Reports*. 2019. vol. 1. no. 4. DOI: 1. 10.1002/eng2.12065.
34. Kordnoori S., Mostafaei H., Kordnoori S., Ostadrahimi M. Testing the Semi Markov Model Using Monte Carlo Simulation Method for Predicting the Network Traffic.

- Pakistan Journal of Statistics and Operation Research. 2020. pp. 713–720. DOI: 10.18187/pjsor.v16i4.3394.
35. FSTEK RF. Metodika ocenki ugroz bezopasnosti informacii [Methodology for assessing information security threats]. Available at: <https://fstec.ru/protivodejstvie-korruptsii/metodicheskie-materialy/metodicheskij-dokument> (accessed: 21.09.2023). (In Russ.).
  36. Voevodin V.A. Monte Carlo method for predicting the stability of the functioning of the informatization object in the conditions of massive computer attacks. Journal of Physics: Conference Series. 2021. vol. 2099. no. 1. DOI: 10.1088/1742-6596/2099/1/012070.

**Voevodin Vladislav** — Ph.D., Associate professor of the department, Department of «information security», National Research University «MIET». Research interests: models, methods and means of auditing the condition of an object under the influence of threats to its information security, models and methods for assessing the security of information, including in the interests of information risk insurance, methods for evaluating the effectiveness of systems (complexes) measures to ensure information security. The number of publications — 60. [vva541@mail.ru](mailto:vva541@mail.ru); 901, Sunny Alley, 124575, Moscow-Zelenograd, Russia; office phone: +7(985)787-1344.

**Acknowledgements.** With the support of the Potanin Foundation.

Д.В. Смирнов, О.О. Евсютин  
**МЕТОДИКА СБОРА ДАННЫХ ОБ АКТИВНОСТИ  
ВРЕДНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОД  
ОС WINDOWS НА БАЗЕ MITRE ATT&CK**

*Смирнов Д.В., Евсютин О.О. Методика сбора данных об активности вредоносного программного обеспечения под ОС Windows на базе MITRE ATT&CK.*

**Аннотация.** Цифровизация современной экономики привела к масштабному проникновению информационных технологий в различные сферы человеческой деятельности. Кроме положительных эффектов это крайне обострило проблему противодействия киберугрозам, реализация которых злоумышленниками часто влечет за собой тяжелые последствия. Вредоносное программное обеспечение (ВПО) занимает важное место на современном ландшафте киберугроз, наиболее громкие киберпреступления последних лет связаны с применением ВПО. В связи с этим активно развивается проблемная область противодействия ВПО и одним из перспективных направлений исследований в данной области является создание методов детектирования ВПО на основе машинного обучения. Однако слабым местом многих известных исследований является построение достоверных наборов данных для моделей машинного обучения, когда авторы не раскрывают особенности формирования, предобработки и разметки данных о ВПО, что компрометирует воспроизводимость этих исследований. В данной работе предлагается методика сбора данных об активности ВПО, основанная на матрице MITRE ATT&CK и Sigma-правилах, и рассчитанная на операционные системы семейства Windows. Предлагаемая методика направлена на повышение качества наборов данных, содержащих характеристики поведения ВПО и легитимных процессов, а также на сокращение времени разметки данных экспертным способом. Для апробации методики подготовлен программный стенд и проведены эксперименты, подтвердившие ее адекватность.

**Ключевые слова:** кибербезопасность, вредоносное программное обеспечение, MITRE ATT&CK, мониторинг активности процессов, машинное обучение.

**1. Введение.** Новое тысячелетие можно охарактеризовать как начало новой вехи научно-технического прогресса. Автоматизация рутинной деятельности привела к тому, что информация стала одним из самых ценных ресурсов XXI века. Ценность информации стала причиной особого интереса к конфиденциальным данным различных спецслужб, подразделений промышленного шпионажа, а также преступников [1]. По этой причине мероприятия по защите информации перешли в отдельную область знаний – информационную безопасность.

Для описания угроз информационной безопасности стали использовать термин «ландшафт киберугроз» [2], который позволяет оценить весь перечень типов актуальных киберугроз для какой-либо отрасли. К примеру, существует ландшафт киберугроз для финансовых учреждений или промышленных предприятий

с использованием киберфизических систем [3]. Попытку реализации киберугрозы назовем атакой [4] или вторжением [5]. Основные типы вторжений были описаны еще в 1987 году в фундаментальной работе [5]. Автор исследования отдельно выделил попытки эксплуатации уязвимостей, атаки внутреннего нарушителя, утечки конфиденциальных данных, реализацию отказа в обслуживании, а также использование вредоносного программного обеспечения (ВПО). Каждая из этих киберугроз направлена на компрометацию одного или более свойств безопасности информации. Однако последний вид угроз является одним из наиболее опасных. В результате реализации атак с использованием ВПО возможно нарушение конфиденциальности, целостности и доступности информации. Поэтому противодействие распространению ВПО за счет создания новых более эффективных методов обнаружения является актуальной проблемной областью.

Существующие методы и средства борьбы с ВПО основываются на изучении известных образцов ВПО с целью написания сигнатур, а также на эвристических способах детектирования наиболее общих признаков, характерных для ВПО. Эти признаки либо извлекаются из файлов вредоносных программ, либо формируются на основе характеристик поведения вредоносных процессов. В соответствии с этим подходы к обнаружению ВПО делятся на статические и динамические.

Для работы с извлеченными признаками в настоящее время достаточно активно стали применяться различные методы машинного обучения. Например, данному вопросу посвящены недавние обзоры [6, 7, 8, 9]. Однако данное направление исследований содержит немало открытых проблем. Большинство исследователей не акцентирует внимание на вопросах формирования выборки ВПО, предобработки и разметки полученных данных. Во многих работах присутствуют ссылки на готовые наборы данных без уточнения деталей формирования этих наборов данных, что компрометирует практическую ценность этих работ. В некоторых исследованиях [10, 11, 12, 13, 14] описывается создание новых наборов данных, однако основной акцент делается на выделении признаков для создания моделей машинного обучения, а вопросы предобработки и разметки полученных данных опускаются. В патентах [15, 16, 17] отечественных и зарубежных лидеров в области кибербезопасности предлагаются отдельные способы формирования паттернов поведения ВПО в виде цепочек событий, однако в качестве метода разметки

данных используется экспертный метод для дальнейшего создания моделей машинного обучения.

Наша работа направлена на повышение качества наборов данных, содержащих характеристики поведения ВПО и легитимных процессов, а также на сокращение времени разметки данных. Под повышением качества понимается обеспечение соответствия набора данных некоторым заранее заданным критериям качества. Наше исследование опирается на критерии, представленные в работе [18]. Краткое описание данных критериев приведено в подразделе 2.2 настоящей статьи. При этом отметим, что разработка подходов к формированию наборов семейств и образцов ВПО и легитимных программ выходит за рамки нашего исследования.

Основным результатом данного исследования является методика сбора данных об активности ВПО под ОС Windows на базе матрицы MITRE ATT&CK [19], где под активностью ВПО понимаются поведенческие признаки вредоносных процессов ОС Windows, создаваемые в ходе работы ВПО. Новизна методики заключается в формализации этапов проведения эксперимента по сбору данных об активности ВПО и требований к содержанию обязательных полей и значений в событиях процессов ОС, а также применении Sigma-правил [20] для автоматизированной разметки этих событий в соответствии с используемыми ВПО техниками, тактиками и процедурами, соответствующими матрице MITRE ATT&CK. В данной работе предлагается использование Sigma-правил для разметки событий, а не для детектирования вредоносной активности. Использование Sigma-правил для обнаружения атак является нецелесообразным ввиду неприемлемого уровня ложноположительных срабатываний на объемной инфраструктуре для всех общедоступных Sigma-правил, а применение ограниченного набора этих правил существенно снизит эффективность обнаружения ВПО [21, 22]. Таким образом, применение предлагаемой методики позволит повысить применимость и воспроизводимость методов обнаружения ВПО на основе машинного обучения.

Статья организована следующим образом. Во втором разделе представлен обзор проблемной области обнаружения ВПО. Третий раздел описывает матрицу MITRE ATT&CK и подход к сопоставлению техник атаки и Sigma-правил. В четвертом разделе рассмотрены механизмы мониторинга активности процессов ОС Windows [23]. Предлагаемая методика сбора данных об активности процессов ОС Windows приведена в пятом разделе. Результаты вычислительных экспериментов с предлагаемой методикой

представлены в шестом разделе. Заключение подводит итоги настоящего исследования.

**2. Обзор подходов к обнаружению ВПО.** Задачу обнаружения ВПО в информационной системе можно формализовать следующим образом. Пусть  $O = \{o_1, o_2, \dots, o_m\}$  – множество объектов-файлов, имеющих в информационной системе, и  $T = \{t_1, t_2, \dots, t_n\}$  – множество моментов дискретного времени. Известно, что множество  $O$  может быть разделено на два подмножества  $O = O_S \cup O_M$ ,  $O_S \cap O_M = \emptyset$ , где подмножество  $O_S$  содержит легитимное ПО, а подмножество  $O_M$  – вредоносное ПО, но неизвестен способ этого разделения.

Множество всевозможных событий, которые могут происходить в информационной системе, назовем алфавитом событий и обозначим  $E = \{e_1, e_2, \dots, e_k\}$ . В алфавит  $E$  входят события файловой системы, сетевые события, события запуска процесса и т.д. В каждый момент времени  $t \in T$  каждому объекту  $o \in O$  может быть поставлена в соответствие последовательность символов  $e^{o,t} = (e_1^{o,t}, e_2^{o,t}, \dots, e_l^{o,t})$ ,  $e_i^{o,t} \in E$ ,  $i = \overline{1, l}$ , которую назовем совокупностью событий, ассоциированных с объектом  $o$  в момент времени  $t$ .

Алфавит событий может быть построен независимо от множества объектов в информационной системе, однако реализация событий, ассоциированных с конкретными объектами, приводит к порождению элементов данных, которые в общем случае различны для различных объектов. Обозначим совокупность элементов данных, порожденных совокупностью событий  $e^{o,t}$  как  $d^{o,t} = (d_1^{o,t}, d_2^{o,t}, \dots, d_l^{o,t})$ ,  $d_i^{o,t} \in A^*$ , где  $A^*$  – множество всевозможных двоичных последовательностей конечной длины. Заметим, что если  $e^{o,t} = \emptyset$ , то и  $d^{o,t} = \emptyset$ .

Тогда задачу обнаружения ВПО в информационной системе определим как построение классификатора следующего вида:

$$F : (o, e^{o,t}, d^{o,t}) \mapsto c \in \{0, 1\}, \quad (1)$$

где  $c = 0$ , если  $o \in O_S$ , и  $c = 1$ , если  $o \in O_M$ .

Данный классификатор извлекает признаковые характеристики анализируемого объекта из совокупности событий, ассоциированных



с данным объектом, и порожденной этими событиями совокупности элементов данных. Очевидно, что при этом время, за которое система обнаружения ВПО выносит решение, должно быть минимизировано.

## 2.1. Извлечение признаков и методы обнаружения ВПО.

На рисунке 1 приведена классификация извлекаемых признаков ВПО, которыми могут оперировать системы обнаружения ВПО [6, 7].



Рис. 1. Классификация извлекаемых признаков ВПО

Первой группой признаков являются статические признаки, которые можно извлечь без запуска файлов. Основной список таких признаков описан в работе [6], содержащей обзор подходов к детектированию исполняемых файлов ВПО с использованием методов интеллектуального анализа данных. Для решения более общей задачи обнаружения ВПО (без привязки к исполняемым файлам) актуализируем способы извлечения основных статических признаков ВПО:

1. Получение уникальных строковых данных [6]. Данный способ извлечения признаков сводится к экспертной оценке аналитиком ВПО уникальности определенных строк, которые являются осмысленными. Например, в файле ВПО присутствует URL-ссылка для загрузки других модулей ВПО с командного центра злоумышленников. Как правило, данная ссылка задается в виде строки без применения методов шифрования или кодирования. С учетом того, что ссылка содержит сетевой адрес сервера злоумышленников и путь к конкретным файлам, маловероятно появление такой строки в файлах, не относящихся к ВПО.

2. Выделение ключевых байт. Данный способ похож на первый, однако в отличие от него вместо уникальных интерпретируемых строк использует байты или их сочетания. Извлечение байтовых признаков происходит от метода общего анализа n-грамм, который широко используется при обработке текстов на

естественных языках [25]. Важной особенностью данного способа является возможность использования в качестве ключевых байт любых данных (в том числе и бинарных данных исполняемого файла).

3. Выделение особенностей опкодов инструкций процессора (Operational Code, OpCode). Данный способ можно назвать частным случаем предыдущего, однако в нем байты представлены в виде исполняемых опкодов. Например, в работе [26] ведется подсчет частоты использования определенных инструкций процессора для легитимных программ и ВПО. Стоит отметить, что этот способ работает только для исполняемых скомпилированных файлов.

4. Разбор структуры файла: извлечение метаданных файла, а также основных структурных признаков в зависимости от типа файла [27]. Анализ структурных признаков файла помогает обнаружить подозрительные типы объектов внутри файлов, в том числе скриптов и неисполняемых документов (например, признаки «AutoOpen» для макросов или OLE-объекты типа «OLE2link»).

5. Изучение особенностей функций исполняемых файлов [28]. Функцией называют последовательность инструкций процессора, которые образуют единую логическую конструкцию с единой точкой входа и одной или несколькими точками выхода. Данная конструкция состоит из базовых блоков (линейных блоков исполняемого кода без инструкций передачи управления) [29]. Особенности функций используют для обнаружения ВПО.

Данный список извлекаемых файловых признаков не является исчерпывающим, но покрывает значительное число основных статических признаков. Стоит отметить, что при извлечении только статических признаков невозможно детектировать определенные виды атак с использованием ВПО: например, бесфайловые атаки [30] или атаки с применением обфусцированных исполняемых образцов [31, 32, 33]. Для решения данной проблемы были разработаны динамические (поведенческие) подходы к детектированию ВПО. Они основаны на извлечении динамических признаков работы процессов ОС, среди которых могут присутствовать процессы ВПО. Поэтому в данном случае для проверки степени вредоносности файла требуется его запуск. К способам извлечения динамических признаков процессов ОС относят [34, 35, 36]:

1. Перехват событий ОС или API-вызовов. Согласно работе [5], к данной категории динамических признаков относятся записи событий аудита. Позже с расширением возможностей по перехвату API-функций ОС и распространением виртуальных машин

для обнаружения ВПО стали использоваться системные вызовы и их аргументы [34].

2. Выделение атрибутов сетевого трафика (например, IP-адреса, DNS-имена, используемые сетевые порты) или байт содержимого трафика. В 1999 году появился первый промышленный стандарт написания сетевых сигнатур и сетевая система обнаружения вторжений (СОВ) Snort [37], основанная на применении регулярных выражений и сочетаний определенных байтов. В данной системе были реализованы принципы работы СОВ, сформулированные в [5].

3. Вычисление метрик использования ресурсов ОС согласно принципам [5] (например, большое количество событий файловой системы может свидетельствовать о работе ВПО типа TrojanRansom [6]).

Как было отмечено ранее, после извлечения признаков для обнаружения ВПО применяются методы детектирования ВПО. Первым наиболее очевидным методом детектирования ВПО является сигнатурный [24], который сводится к сравнению извлеченных значений признаков с сигнатурами [5]. В настоящее время широкое распространение получил общедоступный сканер файлов и одноименный стандарт написания правил детектирования YARA [38, 39]. Большинство вендоров антивирусного ПО при публикации своих исследований используют данный стандарт для обмена информацией об образцах ВПО. Путем декомпозиции сигнатурного метода на несколько первичных [24] выделим следующие методы детектирования ВПО, основанные на работе с определенными классами признаков:

1. Сравнение найденных уникальных строк со строками, характерными для ВПО и собранными в специальной базе строковых сигнатур.

2. Разбиение байт на  $n$ -граммы, построение статистики по  $n$ -граммам и проведение частотного анализа  $n$ -грамм. Преимущество данного метода состоит в том, что он предоставляет возможность сопоставления функционала ВПО и  $n$ -грамм и характеризуется более высокой полнотой обнаружения по сравнению с оценкой частоты встречаемости определенных файловых признаков [25]. Частным случаем данного метода детектирования является метод, основанный на изучении особенностей последовательностей опкодов. Например, в работе [32] проводится выявление определенных последовательностей скомпилированного бинарного кода на основе методов филогенеза, используемых в биологии. Таким образом, в данном подходе к обнаружению ВПО используется сочетание двух способов извлечения статических признаков.

3. Подсчет энтропии разделов файла и их сравнение со значениями, характерными для разделов файлов ВПО. После разбора структуры файла целесообразно рассчитать значение энтропии секции с исполняемым кодом. Если в файле применяется кодирование, то уровень энтропии будет достаточно высоким [31].

4. Графовый анализ. Данный метод применим к сущностям, у которых есть точка входа и точка (или точки) выхода. В случае исполняемых файлов такой сущностью является функция. На основе данного метода строится граф потока управления (Control Flow Graph, CFG) [40], который широко используется для обнаружения исполняемых файлов ВПО, в том числе образцов ВПО с возможностью самомодификации (полиморфное/метаморфное ВПО) [41].

5. Вычисление функций схожести. В данном методе используется сравнение характеристик ВПО с эталонным значением с помощью определенных математических функций. Характерным примером применения данного метода является использование нечетких хешей [33, 41]. Особым направлением исследований стал подход, изложенный в работе [42]. В ней исследователи предложили свести задачу классификации исполняемых файлов к их визуализации в виде изображений, к которым затем применялись методы вычисления функций схожести.

Более подробное описание перечисленных методов детектирования ВПО представлено в обзорных работах [6, 7, 8, 9]. Известно, что сигнатурные методы обнаружения ВПО на основе динамических признаков достаточно широко распространены в системах СОВ, специализированных изолированных средах – песочницах [46], а также решениях класса Endpoint Detection and Response (далее – EDR) [47] на хостах и серверах под управлением ОС Windows. Однако, несмотря на распространенность, указанные методы обладают определенными ограничениями. Необходимость работы в реальном времени предъявляет повышенные требования к производительности работы операционной системы, что сказывается на полноте и точности обнаружения ВПО.

Для решения проблем, которые присущи вышеперечисленным методам детектирования ВПО, стали широко использоваться методы интеллектуального анализа данных. В работе [6] приводится классификация таких методов, используемых в задаче обнаружения ВПО: деревья решений (Decision Tree, DT), байесовские классификаторы (Naive Bayes, NB: наивный байесовский классификатор и мультиномиальный наивный байесовский классификатор). Также за последние 20 лет широкое распространение

получили различные алгоритмы машинного обучения: метод опорных векторов (Support Vector Machine, SVM), метод случайного леса (Random Forest, RF), метод k-ближайших соседей (K-Nearest Neighbor, k-NN), искусственные нейронные сети (Artificial Neural Network, ANN) и глубокое обучение (Deep Learning, DL) [48, 49], различные алгоритмы биоинформатики [50], в том числе и иммунные системы (Artificial Immune System, AIS) [51, 52]. Стоит отдельно выделить методы, в которых совмещены алгоритмы машинного обучения и методы анализа временных рядов [53].

Важным аспектом применения алгоритмов машинного обучения является зависимость их эффективности от качества и объема обучающей и тестовой выборок. Большинство исследователей берут готовые наборы данных для обучения своих моделей. Применительно к задаче обнаружения ВПО такой подход компрометирует практическую ценность соответствующих работ, так как при формировании набора данных по файловым или поведенческим характеристикам ВПО исследователи должны доверять результатам работы других авторов, которые занимались сбором этих данных. Поэтому отдельной задачей для исследователей является создание доверенных наборов данных.

**2.2. Наборы данных для обучения моделей детектирования ВПО.** Наибольшую ценность для исследователей, использующих готовые выборки, представляет работа [18]. В ней авторы исследовали вопросы планирования экспериментов с ВПО. В частности, им удалось выработать критерии качества наборов данных и требования к прозрачности проведения самого эксперимента. К критериям качества наборов данных исследователи отнесли отсутствие в обучающей выборке легитимных файлов, помеченных как ВПО (А.1), сбалансированность выборки (А.2), правильное разделение выборки на обучающую и тестовую (А.3), преобладание привилегий системы сбора данных над привилегиями ВПО (А.4), соответствие среды сбора данных реальным условиям работы алгоритма классификации (А.5) и наличие легитимной фоновой активности (А.6). Таким образом, при использовании готового набора данных исследователи должны убедиться в его соответствии изложенным в работе [18] требованиям А.1–А.6.

Как показывает обзор литературы, большинство известных наборов данных, используемых для обучения моделей детектирования ВПО, не удовлетворяют приведенным требованиям. Кроме того, многие исследователи опускают детали решений, которыми руководствуются при формировании используемых ими наборов

данных. К примеру, в работе [43] использовался набор данных, именуемый «malimg» [54]. Данный набор данных сформирован на основе системы Anubis [55] – одной из первых облачных систем анализа исполняемых файлов. После обработки данных в системе Anubis исследователи произвели отображение бинарных файлов вредоносных программ в наборы 8-битных целых чисел без знака, представленные в виде матриц. Далее полученные матрицы были визуализированы как изображения в градациях серого. Несмотря на достаточно подробное описание алгоритма получения изображения из исполняемого файла, в [43] не обоснован выбор семейств вредоносных файлов для формирования выборки. Описывая набор данных, исследователи упомянули различные алгоритмы сжатия исполняемого кода (например, UPX), но не сопоставили между собой сходства и отличия исследуемых семейств ВПО. Из общего объема выборки (9342 образца ВПО и 25 рассмотренных семейств ВПО) около половины (4540 исполняемых файлов) приходится на 2 подсемейства: Allapple.A и Allapple.L, что нарушает сбалансированность выборки (А.2).

Другим частым случаем нарушения требований к набору данных является отсутствие преобладания привилегий системы сбора данных над привилегиями ВПО (А.4). В основном данное требование характерно для наборов данных, собранных в песочнице Cuckoo Sandbox – проекте с открытым исходным кодом [56]. Это обусловлено архитектурной особенностью данной песочницы: для мониторинга активности исследуемых файлов используются методы API-перехвата в пользовательском режиме. Одним из примеров набора данных, полученного с использованием песочницы Cuckoo Sandbox, является ransomwaredataset2016 [57]. Этот набор данных использовался в работе [35] по автоматизации анализа ВПО класса криптовымогателей (Ransomware), а также в работе [36] по обнаружению данного класса ВПО с использованием алгоритма роя частиц. Другим примером набора данных, полученным с помощью Cuckoo Sandbox, является MalwareDataset [58]. Этот набор данных исследовался в статье [11] в задаче классификации похожих образцов ВПО на основе n-грамм последовательностей вызовов API-функций.

Таким образом, статические и динамические признаки файлов и процессов ОС имеют свои особенности в решении задачи обнаружения ВПО: статические признаки файлов начали использоваться раньше динамических и их использование незначительно сказывалось на работе ОС Windows, на которой производилась антивирусная проверка. Однако использование только статических признаков не позволяет достичь такой же полноты

обнаружения, как и у динамических признаков ВПО, так как существуют полиморфные семейства ВПО и обфускация. По этой причине необходимо использовать динамические признаки процессов ОС Windows, совмещая их с методами машинного обучения для повышения полноты и точности обнаружения ВПО. Для полноценного применения машинного обучения нужны достоверные наборы данных, которые в открытом доступе отсутствуют.

### **3. Модели описания цепочки действий злоумышленника.**

Для классификации ВПО часто используется еще один класс признаков – наличие определенных техник атаки в программном коде (например, сжатие или обфускация исполняемого кода). Для описания техник атаки используются различные модели. Использование этих моделей в работе центра мониторинга кибербезопасности позволяет выстроить процесс выявления активности злоумышленников, а также их атрибуции [59]. Наиболее ранняя модель описания цепочки действий злоумышленника получила название Cyber Kill Chain [60]. Она описывает этапы действий злоумышленника: разведку, подготовку атаки, доставку ВПО, эксплуатацию уязвимости, установку ВПО, получение управления и реализацию риска. Другой моделью для описания вторжений стала Diamond Model [61]. В ней большое внимание уделялось описанию злоумышленника, его возможностей, используемой инфраструктуры и атакуемой жертвы. Однако все эти модели оказались недостаточными для того, чтобы описать наиболее важные аспекты атаки. Например, модель Cyber Kill Chain не конкретизирует действия, происходящие на определенных стадиях атаки, и используемое ВПО. Diamond Model вообще не подразумевает наличие требований к описанию действий злоумышленника. По этой причине в 2016 году организацией MITRE была создана многомерная модель MITRE ATT&CK [19], основой которой являются техники, тактики и процедуры атакующих (далее – TTP: techniques, tactics and procedures). Данная модель регулярно пополняется на основе открытых источников информации, в которых присутствуют описания инструментов атаки и особенностей их использования злоумышленниками.

**3.1. Матрица MITRE ATT&CK.** В настоящее время модель MITRE ATT&CK получила наиболее широкое распространение по сравнению с прочими моделями. Первым исследованием, основанным на применении данной модели, была работа [62]. MITRE ATT&CK является матрицей, представляющей почти готовую методику по классификации признаков присутствия злоумышленников в сети организации на основе определенных тактик, техник и процедур,

реализуемых злоумышленниками. Стоит отметить, что матрица MITRE ATT&CK не является матрицей в математическом смысле. Вероятнее всего, матрицей стали называть двухмерную классификацию действий злоумышленника для удобства восприятия. В MITRE ATT&CK по горизонтали первой строки перечислены тактики, которые являются определенными этапами атаки (они пересекаются с этапами атаки в соответствии с Cyber Kill Chain [59]). Вертикальными ячейками матрицы являются техники – любые значимые действия атакующего. Для решения задачи выявления ВПО важно определение того этапа жизненного цикла атаки, на котором происходит обнаружение вредоносных файлов или подозрительной активности (соответствие принципу минимизации времени инцидента).

Рассмотрим пример инцидента с банковским трояном Emotet [63]. Схема атаки в соответствии с MITRE ATT&CK приведена на рисунке 2.

Данное ВПО является достаточно показательным в контексте рассматриваемой задачи. Приведенная схема охватывает не все техники и тактики атаки, предусмотренные MITRE ATT&CK, но наиболее значимые из них. Важно, что ВПО Emotet используется для доставки ВПО других семейств, то есть является загрузчиком. Обычно первыми двумя этапами атаки считаются разведка (Reconnaissance) и подготовка инфраструктуры (Resource Development). Однако они происходят на стороне злоумышленника и по этой причине для детектирования ВПО интереса не представляют, поэтому в отчете об инциденте они не приведены. Первый видимый на стороне атакуемой инфраструктуры этап первоначального доступа (Initial Access) производился с помощью рассылки писем, содержащих вредоносный LNK-файл (техника «Phishing»). Следующие два этапа выполнение (Execution) и закрепление (Persistence) позволяют ВПО запуститься и закрепиться в системе, поэтому первые три этапа на приведенной выше схеме важны для обнаружения и предотвращения дальнейшей работы ВПО. Обычно их старается предотвратить большинство антивирусных программ, перехватывая событие создания нового процесса и сканируя запускаемый образ. Далее идут этапы повышения привилегий (Privilege Escalation) и предотвращение обнаружения (Defense Evasion), на которых злоумышленник повышает свои привилегии и противодействует средствам детектирования. В данном случае была использована техника внедрения кода (Process Injection) в процесс Winlogon, которая заключается в запуске вредоносного кода в контексте другого процесса для обхода детектирования и повышения привилегий для работы от системной учетной записи. Данный пример демонстрирует, что иногда техника позволяет добиться сразу двух



целей: в таком случае эта техника попадет сразу в две колонки, соответствующие тактикам.

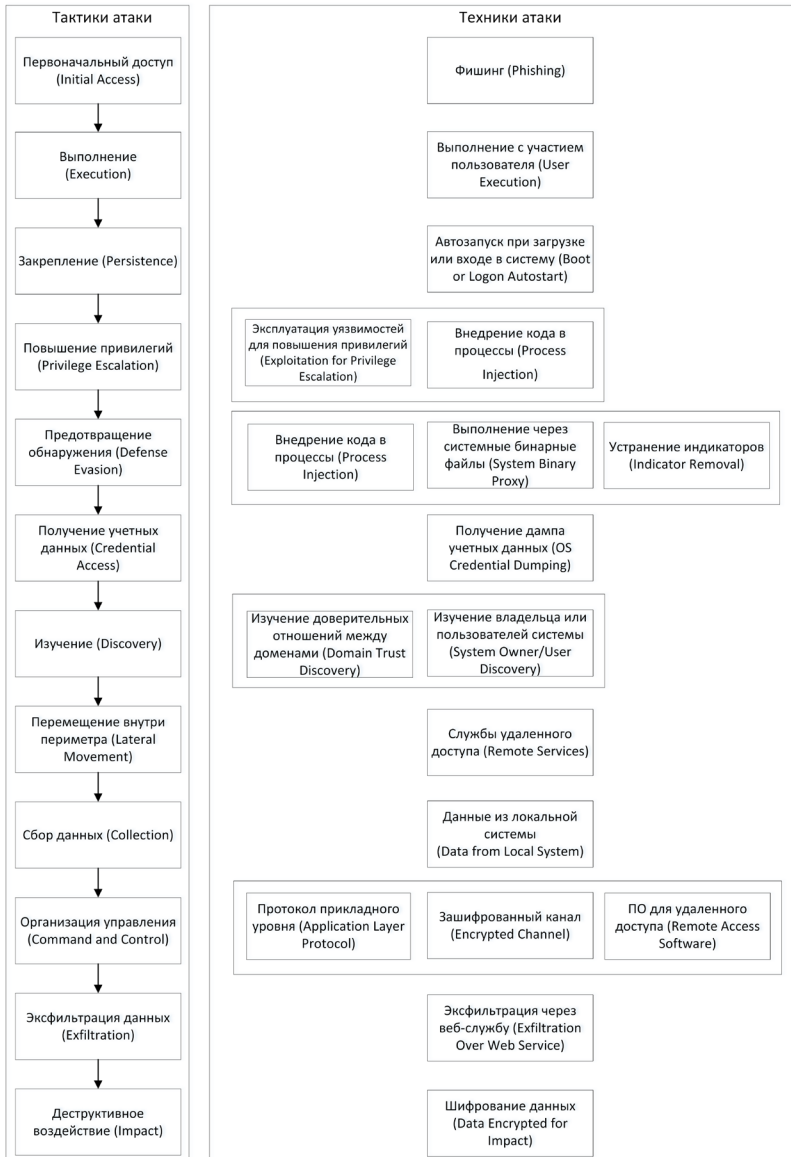


Рис. 2. Пример схемы атаки в соответствии с MITRE ATT&CK

После этих действий обычно наступают дальнейшие изучение (Discovery), захват корпоративной сети (Credential Access, Lateral Movement, Command and Control) и деструктивное воздействие (Impact). На примере инцидента с ВПО Emotet в конечном счете было запущено вымогательское ВПО Quantum. Если злоумышленник смог пройти данные этапы, это значит, что он смог скомпрометировать компьютерную сеть.

Для обнаружения ВПО на определенных этапах атаки требуется механизм сопоставления признаков проявления ВПО с данными из заранее собранной базы. Исследователи из компании Mandiant [64] создали такую базу сигнатур и специальный инструмент *capa* [65], который выводит список используемых в исполняемом коде ВПО техник атаки в соответствии с MITRE ATT&CK. Однако данный подход имеет существенные ограничения ввиду того, что он работает только для исполняемых файлов.

**3.2. SIEM-системы и Sigma-правила.** Как было отмечено ранее, подходы к детектированию ВПО на основе статических признаков имеют существенные ограничения в виде невозможности обнаружения бесфайлового ВПО и обфусцированных вредоносных программ. В настоящее время в корпоративных сетях используется централизованный подход к получению данных о поведенческих особенностях работы клиентских и серверных систем. На узлах сети помимо антивирусного решения зачастую используется хостовый агент сбора системных событий в виде EDR-решения [47]. Полученные события записываются в журналы, которые собираются и централизованно обрабатываются в SIEM-системе (Security Information and Event Management – управление событиями и информацией о безопасности) [66]. Данная система служит для нормализации, агрегации и корреляции событий с различных источников корпоративной сети (например, антивирусов, межсетевых экранов, средств обнаружения вторжений). Корреляция событий происходит по специальным правилам, которые являются поведенческими сигнатурами [66].

Большинство SIEM-систем имеют свой стандарт описания правил корреляции. В настоящее время существует проект Sigma [20], направленный на унификацию поисковых запросов и соответствующих правил корреляции для популярных SIEM-систем. Синтаксис правил Sigma соответствует логике правил обнаружения вторжений, описанных в [5]. Важным параметром качества правил является степень покрытия матрицы MITRE ATT&CK этими правилами [67]. В рамках текущего исследования сформулирована

следующая гипотеза: сбор данных об активности ВПО можно автоматизировать путем использования средства сбора событий активности процессов ОС, а журналы работы данного средства целесообразно обрабатывать с помощью правил, поддерживающих синтаксис Sigma и привязку к MITRE ATT&CK. Таким образом, формирование набора данных об активности ВПО возможно путем сбора событий активности процессов ОС и последующей их обработки с использованием Sigma-правил, позволяющих построить соответствие «событие ОС → техника атаки».

**4. Механизмы мониторинга активности процессов ОС Windows.** Динамические признаки позволяют добиться более высокой эффективности обнаружения ВПО по сравнению со статическими. В качестве источников данных в ОС Windows для извлечения динамических признаков используются различные категории событий:

1. События, генерируемые на основе перехвата API-функций (API Hooking) [68]. Данный механизм является наиболее информативным для задач кибербезопасности, однако не относится к штатным: его применение в качестве основного механизма мониторинга активности процессов ОС Windows несет определенные проблемы. Во-первых, такой проблемой является зависимость от версии сборки ОС Windows в случае использования перехвата API-функций в режиме ядра [23]. Во-вторых, данный механизм уязвим к атакам типа «обход защиты» (Defense Evasion согласно MITRE ATT&CK [19]) в случае использования перехвата API-функций в пользовательском режиме [23].

2. События, генерируемые агентом или внутренним механизмом гипервизора песочницы для поведенческого анализа файлов. Наиболее известным общедоступным проектом, в котором реализован данный механизм, является песочница Cuckoo Sandbox [56]. Другим популярным общедоступным решением является drakvuf [69]. Однако такой подход к мониторингу активности процессов ОС Windows возможен только на виртуальных машинах или гипервизорах.

3. События, генерируемые специализированными драйверами с определенными функциями обратного вызова (например, драйверы-минифильтры файловой системы). Широко используются в антивирусных программах, так как данный механизм расширяет возможности по сбору событий и позволяет производить активное реагирование [70]. Имеет существенный недостаток: предоставляет информацию по ограниченному количеству API-функций.

4. События ОС Windows из стандартных журналов System, Security, PowerShell и других [5]. Для использования данных журналов необходимо включить определенные типы событий аудита ОС. Однако эти журналы являются недостаточно информативными для задач кибербезопасности. При этом необходимо отметить, что события ОС Windows в общем случае записываются в журналы благодаря механизму Event Tracing for Windows [71]. Данный механизм является гораздо более гибким, но в то же время сложным и слабо документированным. В настоящее время активно изучается исследователями кибербезопасности, но пока используется недостаточно широко.

Последние два механизма реализованы в бесплатной утилите Sysmon [72] из набора Sysinternals. Данная утилита не является полнофункциональным инструментом аналитики, а лишь предоставляет возможность записи определенных событий в соответствующий журнал. Перечень фиксируемых событий задается специальным конфигурационным файлом перед установкой утилиты в качестве службы ОС Windows. Кроме того, имеется возможность журналировать только те события, которые соответствуют определенным условиям, задаваемым аналитиком.

**5. Методика сбора и разметки данных об активности процессов ОС Windows.** Несмотря на попытки сбора данных об активности вредоносных программ с использованием утилиты Sysmon и публикации соответствующего общедоступного набора данных [73], исследователи до сих пор не создали в открытом доступе набор размеченных цепочек событий, соответствующих ВПО и легитимным файлам. В данном исследовании предлагается использовать вышеперечисленные механизмы мониторинга активности процессов ОС Windows в составе оригинальной методики формирования воспроизводимого набора данных об активности процессов ОС Windows с возможностью дальнейшей автоматизации разметки данных на основе сигнатур Sigma и матрицы MITRE ATT&CK. Таким образом, решается проблема обеспечения достоверности сбора данных о поведенческих признаках ВПО.

Представим формальное описание решаемой задачи с использованием обозначений, введенных в разделе 2. Пусть  $O = \{o_1, o_2, \dots, o_m\}$  – множество объектов-файлов, имеющих в информационной системе, и  $T = \{t_1, t_2, \dots, t_n\}$  – множество моментов дискретного времени. Рассмотрим множество совокупностей событий, ассоциированных с объектами из множества  $O$  в момент времени  $t$ ,

и множество совокупностей порожденных этими событиями элементов данных:  $e^{o_1,t}, d^{o_1,t}, e^{o_2,t}, d^{o_2,t}, \dots, e^{o_m,t}, d^{o_m,t}$ . Некоторым образом осуществим разбиение данных совокупностей на составляющие

$$g_{o'=\{o_1, \dots, o_p\}} = \left( \dots, \left( e_j^{o_j,t}, d_j^{o_j,t} \right), \dots, \left( e_q^{o_p,t}, d_q^{o_p,t} \right), \dots \right),$$

включающие в себя события, ассоциированные с разными объектами из множества  $O$ , и соответствующие данные. Данные составляющие назовем цепочками событий. Тогда задача формирования набора данных об активности вредоносных программ в информационной системе будет состоять в построении классификатора следующего вида:

$$V : g_{o'=\{o_1, \dots, o_p\}} = \left( \dots, \left( e_j^{o_j,t}, d_j^{o_j,t} \right), \dots, \left( e_q^{o_p,t}, d_q^{o_p,t} \right), \dots \right) \mapsto c \in \{0, 1\}, \quad (2)$$

где  $c = 0$ , если  $O' \cap O_S = \emptyset$ , и  $c = 1$  в противном случае.

Необходимо отметить, что отличие от задачи обнаружения ВПО заключается в том, что классификатор  $V$  не относит отдельно взятые объекты к вредоносному или легитимному ПО, а определяет наличие или отсутствие вредоносной активности в совокупности событий, ассоциированных с некоторым подмножеством множества объектов  $O$ .

Как было отмечено ранее, для сбора данных об активности ВПО целесообразно использовать средства виртуализации для изоляции вредоносного воздействия на реальные вычислительные ресурсы и сетевое окружение. Для этого необходимо подготовить виртуальную машину, в том числе произвести установку типичной программной среды пользователя (редакторы офисных файлов, браузеры, файловые менеджеры, архиваторы), настройку сетевых интерфейсов и снятие снимка состояния [74]. Также необходимо произвести установку и конфигурацию специализированных программных средств для сбора данных об активности процессов ОС Windows.

Таким образом, решение поставленной задачи применительно к ОС Windows может быть представлено в виде методики, включающей четыре основных этапа:

1. Управление состояниями VM.
2. Запись журналов активности процессов ОС Windows.
3. Сбор журналов активности процессов ОС Windows.
4. Хранение и обработка журналов.

Приведенные этапы целесообразно выполнять с использованием соответствующих программных средств, как это показано на

рисунке 3. На данной схеме слева показаны этапы, выполняемые на стороне хостовой ОС, а справа – этапы, выполняемые внутри гостевых ОС виртуальных машин.

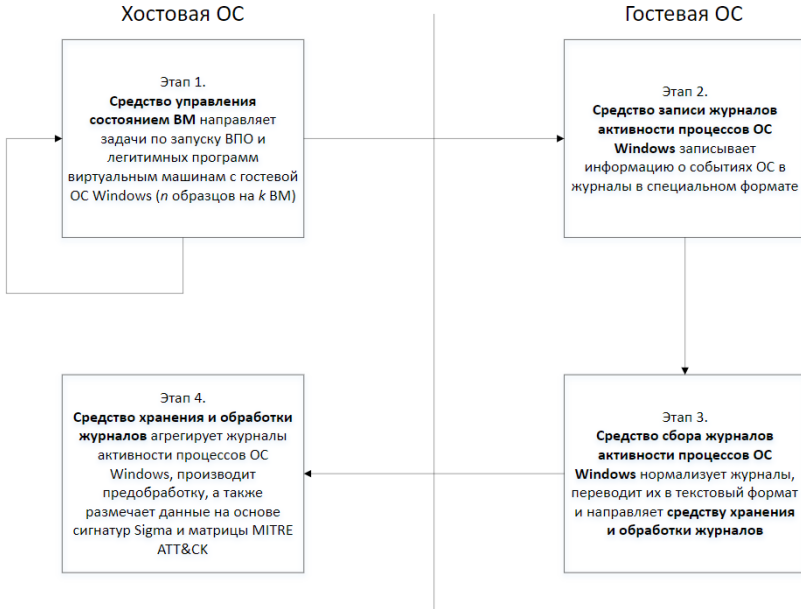


Рис. 3. Методика сбора и разметки данных об активности процессов ОС Windows

Первые три этапа предполагают проведение вычислительного эксперимента по запуску ВПО и легитимных программ для сбора данных об их активности. Последний этап необходим для разметки событий активности исследуемых процессов. Далее рассмотрим этапы предлагаемой методики более подробно.

**5.1. Сбор данных об активности ВПО и легитимных программ.** На первом этапе сбора данных об активности процессов ОС Windows производится распределение задач по запуску ВПО и легитимных программ по виртуальным машинам. Для этого используется программное средство управления состоянием виртуальных машин и вызова файлов ВПО и легитимных программ (рисунок 4). Поскольку в реальных сценариях реализации киберугроз с использованием ВПО запуск ВПО может произойти в любой момент времени после начала работы системы, то при формировании задач

по запуску ВПО и легитимных программ на виртуальных машинах время запуска должно выбираться случайным образом.

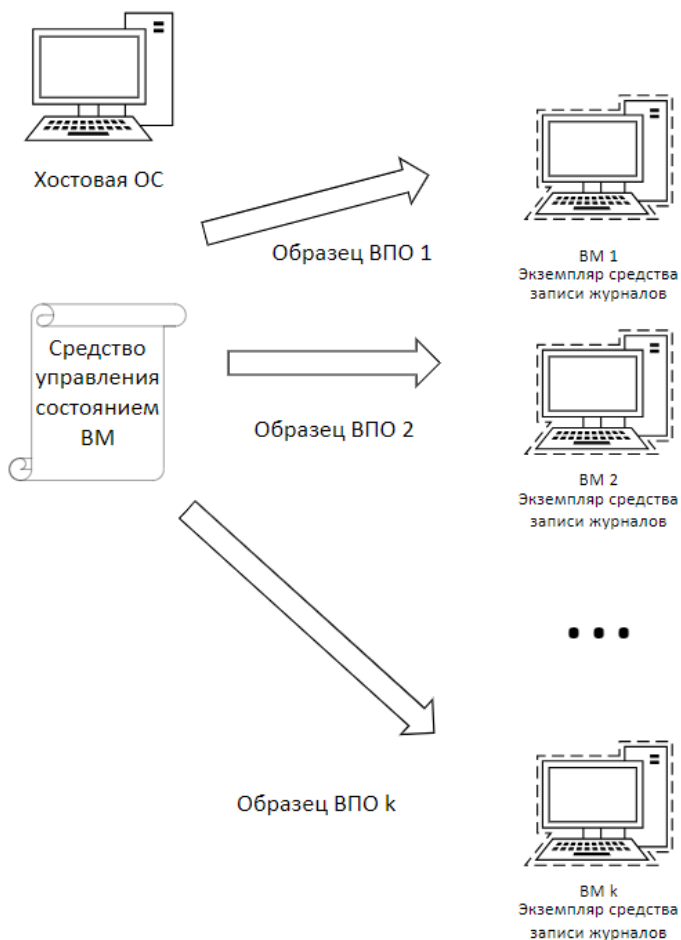


Рис. 4. Распределение задач по анализу образца ВПО и легитимных программ между виртуальными машинами

На втором этапе сбора данных об активности процессов ОС Windows осуществляется запись событий ОС в журналы. Для этого требуется заранее произвести конфигурацию средства записи журналов активности процессов ОС Windows, чтобы обеспечить мониторинг необходимых событий и исключить наименее важные

с точки зрения кибербезопасности события. Необходимые поля событий приведены в разделе 5.2, а пример конфигурации описан в разделе 6.1.

На третьем этапе производится синтаксический разбор, нормализация и пересылка журналов средству хранения и обработки журналов. По истечении определенного для каждой виртуальной машины таймаута средство управления состоянием виртуальных машин должно остановить выполнение текущей задачи, выполнить откат конкретной VM к сделанному на этапе подготовки эксперимента снимку состояния и направить новую задачу. По этой причине средство управления состоянием виртуальных машин должно работать асинхронно для каждой VM. Описанные этапы методики схематично изображены на рисунке 5.

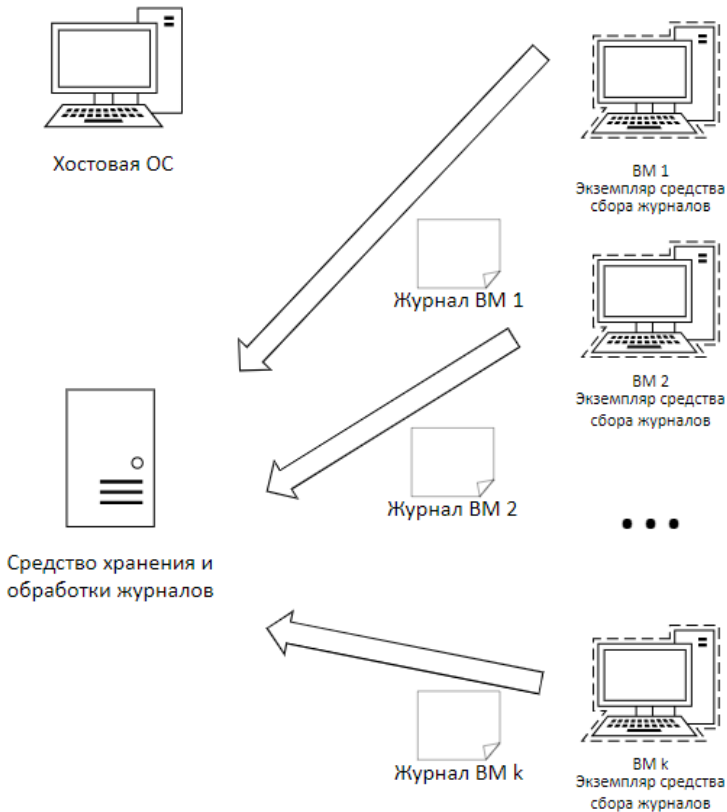


Рис. 5. Сбор и обработка журналов активности процессов ОС Windows



**5.2. Разметка событий активности процессов ВПО и легитимных программ.** Как было отмечено ранее, четвертый этап работы методики относится к разметке событий активности процессов ВПО и легитимных программ на основе Sigma-правил и матрицы MITRE ATT&CK. До проведения разметки данных необходимо проверить корректность и непротиворечивость проверяемых событий и используемых Sigma-правил, в том числе и на отсутствие синтаксических ошибок. Для проверки записанных журналов активности процессов ОС Windows сигнатурами Sigma необходимо соблюдать соответствие полей записанных событий и полей, используемых в Sigma-правилах. Поля записи событий активности процессов ОС Windows, необходимые для корректной работы Sigma, приведены в таблице 1.

Таблица 1. Поля событий, необходимые для проверки Sigma-правилами

Содержимое поля события	Назначение поля события
Тип и название события	Идентификация действия
Имя и сессия пользователя	Идентификация пользователя, от имени которого работает процесс ОС Windows
Идентификатор процесса, полное имя файла и хеш исполняемого образа процесса	Идентификация экземпляра и образа процесса
Идентификатор родительского процесса и полное имя файла образа родительского процесса	Идентификация экземпляра и образа родительского процесса
Командная строка, с помощью которой был создан данный процесс ОС Windows	Идентификация условий, при которых был запущен процесс
Идентификатор и имя объекта, над которым выполняется действие	Идентификация объекта
Запрашиваемые права для действия над объектом	Идентификация запрашиваемых прав доступа процесса к объекту
Дополнительные поля	Любые дополнительные поля, специфичные для определенных действий (например, адрес памяти старта потока для события CreateRemoteThread)

Приведенная таблица позволяет корректно сконфигурировать средство записи журналов активности процессов ОС Windows перед запуском вычислительных экспериментов и затем произвести

проверку записанных событий Sigma-правилами. После указанной проверки необходимо обогатить события новыми полями, содержащими информацию о результатах проверки событий соответствующими Sigma-правилами и приведенными в таблице 2. По умолчанию данные поля должны быть предзаполнены пустыми значениями, в которые вносятся изменения только в случае положительного результата проверки события хотя бы одним Sigma-правилом.

Таблица 2. Новые поля событий по итогам разметки

Содержимое поля события	Назначение поля события
Булево значение, если исполняемый образ процесса соответствует запускаемому образцу ВПО (для запускаемых легитимных программ равно 0, для образцов ВПО равно 1)	Идентификация реальных техник атаки злоумышленника с использованием ВПО
Идентификаторы техник атаки на основе метаданных Sigma-правил	Идентификация потенциальных техник атаки злоумышленника
Имена Sigma-правил	Идентификация Sigma-правил

Разметка считается успешно выполненной, если для всех событий заполнены все поля из приведенной таблицы.

Данная методика работы в целом соответствует логике работы Cuckoo Sandbox, однако в оригинальной методике предлагается использовать штатные механизмы мониторинга активности процессов ОС Windows и применять унифицированные Sigma-правила для обнаружения ВПО, а не специфичные для функционирования Cuckoo Sandbox скрипты [75]. Таким образом, предлагаемая методика позволяет различным исследователям самостоятельно формировать наборы данных по активности ВПО и использовать собственные или общедоступные Sigma-правила для разметки этих данных. Стоит отметить, что репозиторий Sigma-правил постоянно пополняется новыми правилами, а значит новые техники атаки могут быть обнаружены с помощью данных правил.

**6. Проведение экспериментов по сбору данных об активности процессов ОС Windows.** Для проведения экспериментов с предлагаемой методикой была развернута и подготовлена виртуальная среда на базе VirtualBox с двумя гостевыми ОС Windows 10. С учетом динамического характера методики

целесообразно учесть определенные ограничения. На результаты экспериментов с запуском ВПО могут влиять особенности изолированной среды: время работы гостевых ОС и степень их загруженности. По этой причине необходимо производить запуск одного файла на нескольких гостевых ОС. Однако с учетом достаточно высоких требований к работе современных ОС Windows на среднем ПК с 8 Гбайт ОЗУ и 4 ядрами процессора возможно запустить не более двух гостевых ОС без существенной потери производительности хостовой ОС. Для снижения влияния временных характеристик на результаты экспериментов были использованы две идентичные гостевые ОС Windows 10, на которые было выделено разное количество ресурсов хоста: на виртуальную машину VM<sub>1</sub> было выделено 2 ядра процессора, 4 Гбайт ОЗУ и 70 Гбайт ПЗУ, а на виртуальную машину VM<sub>2</sub> – одно процессорное ядро, 2 Гбайт ОЗУ и 30 Гбайт ПЗУ.

Для управления состояниями виртуальных машин и распределения задач по запуску ВПО было разработано отдельное программное средство на языке Python 3 с использованием библиотеки *virtualbox*. В разработанном программном средстве реализован функционал выполнения команд по запуску файлов ВПО на виртуальных машинах и откат этих виртуальных машин к первоначальному состоянию по таймауту.

**6.1. Подготовка виртуальной среды к проведению экспериментов.** В качестве средства записи журналов активности процессов ОС Windows целесообразно выбрать инструмент Sysmon, так как он использует штатные механизмы ОС Windows для перехвата системных событий на уровне ядра ОС Windows и отличается наибольшей гибкостью конфигурирования. Для проведения экспериментов конфигурация службы Sysmon настроена следующим образом: включен сбор всех возможных событий, кроме событий с EventID 27 (блокировка создания исполняемых PE-файлов), так как данные события не только отслеживаются, но и блокируются, что неприемлемо для проведения экспериментов. Дополнительно из мониторинга исключаются события, сгенерированные необходимыми для экспериментов утилитами Sysmon и Elastic Agent и процессом виртуальной машины VirtualBox. Конфигурация Sysmon, использованная в настоящем исследовании, опубликована на GitHub [76].

Для хранения и обработки журналов активности процессов ОС Windows рекомендуется использовать любую SIEM-систему или систему обработки больших данных. Бесплатным программным

средством обработки больших данных является ELK-стек, включающий в себя компоненты Elasticsearch, Logstash и Kibana [77]. В рамках экспериментов за основу была взята и адаптирована конфигурация для инсталляции в контейнере docker-elk [78].

Для автоматизированного сбора журналов активности процессов ОС Windows целесообразно использовать готовый программный инструмент Elastic Agent, который имеет API-интеграцию с ELK-стеком [79]. Данное программное средство позволяет производить парсинг журналов событий Sysmon, осуществлять нормализацию данных и пересылать информацию по сети на веб-сервис Elasticsearch. Файл конфигурации для данной утилиты можно сгенерировать из самой веб-панели управления Kibana и затем установить его на виртуальных машинах для запуска ВПО.

Далее необходимо производить разметку полученных событий Sysmon с помощью Sigma-правил. Для этого была установлена утилита Sigma-cli, позволяющая переводить Sigma-правила в запросы для ELK [80]. Помимо возможности выявления вредоносных событий Sysmon под ОС Windows на базе стека ELK, данная утилита поддерживает возможность перевода Sigma-правил для обнаружения аномальной сетевой активности, подозрительных событий ОС Linux и ОС macOS для систем типа Splunk и большинства распространенных SIEM-решений.

Полный перечень установленного ПО приведен в таблице 3.

Таблица 3. Установленное ПО для проведения экспериментов

Класс ПО	Локация	Название ПО	Версия
Хостовая ОС	Хост	Windows 10	21H2 19044.2486
Средство виртуализации	Хост	VirtualBox	6.1.46 r158378
Средство управления состоянием ВМ	Хост	Python3 virtualbox (библиотека python)	Python 3.10.2 virtualbox 2.1.1
Гостевые ОС	Виртуальная машина (для запуска ВПО)	Windows 10 (для ВМ <sub>1</sub> и ВМ <sub>2</sub> )	21H2 19044.2486
Офисное ПО	Виртуальная машина (для запуска ВПО)	Microsoft Office	2007
ПО для просмотра PDF файлов	Виртуальная машина (для запуска ВПО)	Adobe Reader	8
Браузер	Виртуальная машина (для запуска ВПО)	Internet Explorer	8

Класс ПО	Локация	Название ПО	Версия
Средство записи системных журналов	Виртуальная машина (для запуска ВПО)	Sysmon Sysinternals	15, версия схемы 4.90
Средство сбора журналов	Виртуальная машина (для запуска ВПО)	Elastic Agent	8.1.1
Гостевая ОС	Виртуальная машина (для развертывания ELK-стека)	Ubuntu	22.04.2 LTS Desktop
Средство контейнеризации	Виртуальная машина (для развертывания ELK-стека)	Docker	24.0.4
Средство хранения и обработки журналов	Виртуальная машина (для развертывания ELK-стека)	ELK стек	8.1.1
Средство преобразования Sigma-правил в запросы к Elasticsearch	Виртуальная машина (для развертывания ELK-стека)	Sigma-cli	0.7.6
Sigma-правила	Виртуальная машина (для развертывания ELK-стека)	Sigma	Merge pull request #4353

**6.2. Формирование выборки исследуемого ПО.** В рамках проведения экспериментов были выбраны семейства ВПО [81], приведенные в таблице 4.

Критерием отбора было наличие антивирусных исследований, опубликованных в открытых источниках. Из данных семейств были отобраны 100 образцов ВПО, особенности поведения которых хорошо известны, что позволяет достоверно оценивать корректность разметки данных в ходе экспериментов. Также в таблице приведены классы легитимного ПО, которое является достаточно распространенным на компьютерах пользователей и имеет определенные поведенческие сходства с ВПО (например, средства для шифрования файлов пользователя осуществляют такие же действия, как и вымогательское ВПО класса шифровальщиков).

Таблица 4. Исследуемые экземпляры ВПО и легитимных программ

Тип ПО	Класс ПО	Семейства	Кол-во образцов
ВПО	ВПО, используемое АРТ-группировками (эксплойты, трояны-загрузчики, средства удаленного управления)	Bisonal, MustangPanda, NetTraveler, PittyTiger, Taidoor, Winnti	23
	ВПО, используемое финансово мотивированными группировками (эксплойты, трояны-загрузчики, средства удаленного управления, программы-вымогатели)	CobaltGang, Silence, TA505 (Amadey), RTM, FIN7, Lazarus	25
	Средства удаленного управления	Darkcomet, AsyncRAT, Remcos RAT	5
	Средства кражи паролей и другой конфиденциальной информации	Agent Tesla, Lokibot, IcedID, Formbook	10
	Трояны-загрузчики	Zloader, Emotet, Trickbot, Qbot	6
	Трояны-майнеры	LemonDuck, Monero (XMR) Miner	5
	Программы-вымогатели	WastedLocker, GandCrab, модифицированный WannaCry (была удалена проверка регистрации домена), NotPetya, Egregor, REvil, BlackMatter, Ryuk, DoppelPaymer	14
Хакерские инструменты	Средство извлечения паролей из памяти ОС, фреймворки удаленного управления	Mimikatz, Cobalt Strike, Meterpreter	3
Легитимное ПО	Редакторы (офисное ПО), браузеры, архиваторы, программы шифрования, антивирусные программы	Microsoft Office Word, Microsoft Office Excel, Adobe Reader, Notepad++, Google Chrome, WinRAR, 7zip, VeraCrypt, Антивирус Windows Defender	9

### 6.3. Проведение серии экспериментов и анализ результатов.

Для апробации методики была выполнена серия экспериментов

по запуску перечисленных образцов ВПО и легитимных программ, представленных в таблице 2.

*Эксперимент 1 «Выбор таймаута для атомарного исследования активности ПО».* В ходе первого эксперимента над одним исследуемым образцом ВПО был выбран таймаут, равный одной минуте в соответствии с временем по умолчанию для проверки одного файла в песочнице Cuckoo Sandbox [82]. Данный эксперимент показал, что работа вредоносной программы завершилась мгновенно (между событиями создания и завершения процесса прошло менее одной секунды) ввиду отсутствия доступа к управляющему серверу ВПО. Даже при успешном соединении с управляющим сервером большинство вредоносных программ обрабатывало достаточно быстро: менее 15 секунд при Интернет-канале 20 Мбит/с. Исключениями были два типа вредоносных программ: загрузчики ВПО и программы-вымогатели. Загрузчики ВПО активировались через какое-то время или переходили в режим ожидания команд от злоумышленников. Программы-вымогатели, в свою очередь, продолжают работать до тех пор, пока не зашифруют все целевые файлы в системе. Таким образом, для всех программ был установлен таймаут 30 секунд. Данное значение обусловлено тем, что за указанное время вредоносные события с виртуальной машины попадут в ELK, а для ВПО с отложенным запуском достаточно выявлять событие создания запланированной задачи для дальнейшего запуска.

*Эксперимент 2 «Запуск исследуемых программ».* На обеих виртуальных машинах поочередно запускались образцы ВПО и легитимных программ, представленные в таблице 2. На каждой машине с заданной конфигурацией Sysmon за 30 секунд генерировалось от 2300 до 7800 событий Sysmon, при этом объем выделенных ресурсов на разные виртуальные машины незначительно влиял на количество генерируемых событий. Таким образом, предположение о влиянии количества ресурсов и быстродействия системы на генерацию событий оказалось ошибочным.

Далее был произведен анализ распределения событий Sysmon в зависимости от типов событий. Значительную долю этих событий составляли события взаимодействия с реестром, загрузки исполняемого образа и доступа одного процесса к памяти другого процесса. Для примера приведено распределение типов событий для класса ВПО «Средства удаленного управления» (рисунок 6). Данное распределение является характерным для большинства семейств ВПО и легитимных программ.

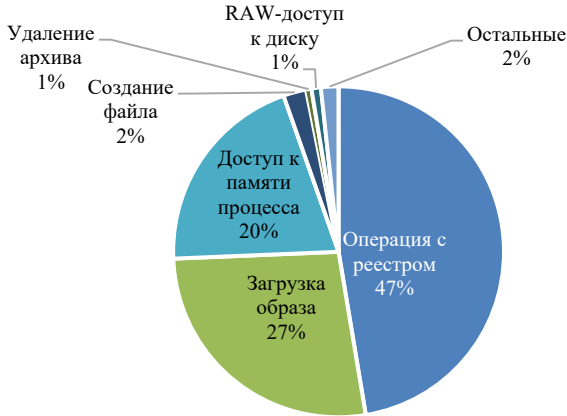


Рис. 6. Распределение событий Sysmon в зависимости от типов событий для класса ВПО «Средства удаленного управления»

Совершенно другая круговая диаграмма распределения характерна для программ-вымогателей: особенностью данного класса ВПО является наличие повышенной файловой активности (рисунок 7).

Из похожести приведенных диаграмм распределения событий Sysmon в зависимости от типов событий для большинства семейств ВПО и легитимных программ следует вывод: большинство зафиксированных событий являются характерными для фоновой активности системы. Таким образом, необходимо провести отдельный эксперимент по исследованию фоновой активности ОС Windows.

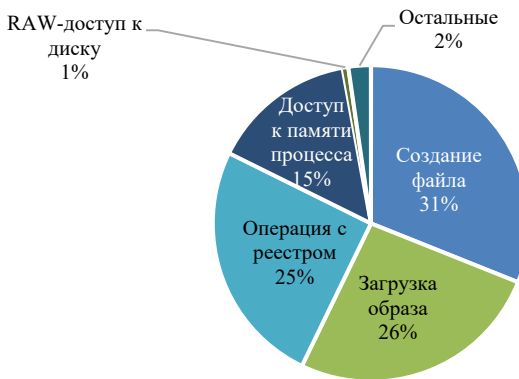


Рис. 7. Распределение событий Sysmon в зависимости от типов событий для класса ВПО «Программы-вымогатели»



*Эксперимент 3 «Исследование фоновой активности ОС Windows».* Для исследования особенностей фоновой активности был проведен следующий эксперимент, направленный на сравнение количества событий на этапе запуска ОС Windows и во время ее работы. Для этого были проведены запуски обеих виртуальных машин с гостевой ОС Windows 10 и зафиксирована работа в течение 12 минут без выполнения программ из таблицы 2. Распределение количества событий по минутам для ВМ<sub>1</sub> приведено на рисунке 8.

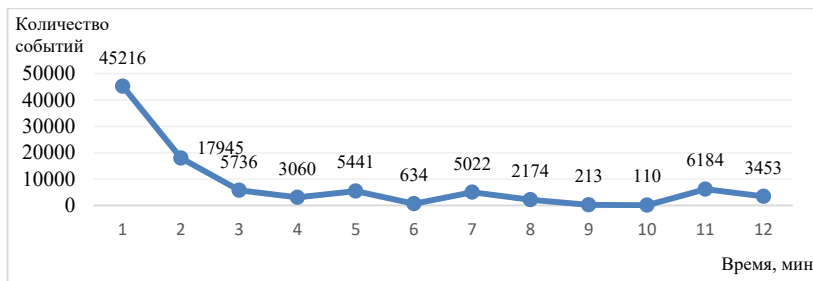


Рис. 8. Количество событий Sysmon после старта ОС Windows 10 в единицу времени

Как видно из данного графика, через 2 минуты после старта системы количество событий становится достаточно равномерным. Разброс значений от минимального 110 до максимального 6184 объясняется особенностью работы самой ОС Windows: возникновение одной фоновой задачи генерирует сразу несколько сотен событий. Таким образом, лавинообразные значения количества событий на графике являются естественными для данной ОС.

Также рисунок 8 подтверждает то, что значительная доля событий Sysmon генерируется самой ОС Windows 10 в фоновом режиме: количество событий Sysmon через 2 минуты после старта ОС Windows входит в интервал количества событий из эксперимента 2. Далее был произведен расчет числа событий Sysmon в зависимости от типов событий для фоновой активности ОС Windows (рисунок 9).

Таким образом, можно модифицировать конфигурационный файл Sysmon, добавив в него значительное количество исключений, характерных для фоновой активности ОС Windows. Модифицированный конфигурационный файл Sysmon доступен в Github авторов данной статьи [76].

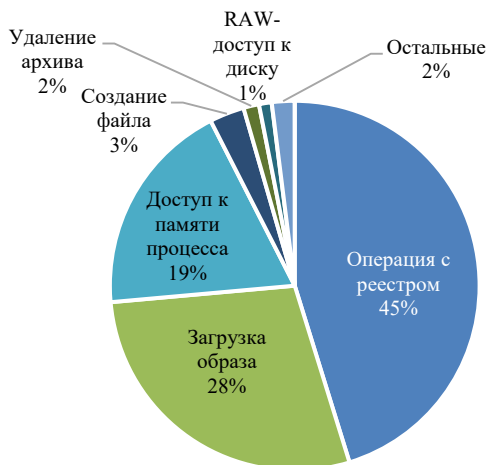


Рис. 9. Распределение событий Sysmon в зависимости от типов событий фоновой активности ОС Windows

*Эксперимент 4 «Выявление сработавших Sigma-правил».* По итогам эксперимента 2 был произведен сбор событий, генерируемых исследуемыми образцами ВПО и легитимных программ из таблицы 2. Для разметки полученного набора данных были запущены утилита Sigma-cli и соответствующий процесс трансляции Sigma-правил в запросы для ELK. По итогу работы данной утилиты был сгенерирован 1641 запрос для ELK. Полученные запросы доступны в Github авторов данной статьи [76].

Далее полученные запросы направлялись в ELK для поиска событий, удовлетворяющих вредоносным шаблонам поведения. В ходе запуска данных запросов было установлено, что 790 из них имеют синтаксические и логические ошибки, связанные с экранированием различных символов: пробелов, двоеточий, слешей и других управляющих символов. Однако с учетом того, что все правила транслируются по единому алгоритму, вероятно, устранение нескольких ошибок в трансляторе сделает все запросы к ELK валидными.

По итогам обработки всех запросов в ELK было установлено, что для 83 из 100 образцов ВПО было обнаружено хотя бы одно вредоносное событие, для 40 образцов – два вредоносных события и для 9 образцов ВПО – три вредоносных события. Для поисковых запросов к ELK и обнаруживаемым ими событиям справедливо утверждение, что одно событие обнаруживается не более чем одним

поисковым запросом. Таким образом, для разметки данных для каждого события были добавлены 5 полей: «sigma\_name» и «sigma\_id» для идентификации Sigma-правил и поисковых запросов к ELK, с помощью которых было задетектировано то или иное событие; «technique\_name» и «technique\_id» для идентификации соответствующих техник атаки; а также поле «is\_malware» – в случае, если данное событие было сгенерировано запускаемым образцом ВПО. Таким образом, 83% исследованных образцов ВПО были справедливо идентифицированы как ВПО. Среди событий, сгенерированных в ходе работы легитимного ПО, не было обнаружено вредоносных, то есть ошибок первого рода не допущено.

Таблица 5. Идентифицированные техники ВПО

Техника атаки, обнаруженная с помощью Sigma-правила	Количество обнаруженных образцов ВПО
Закрепление в системе	71
Обход средств защиты (запуск через легитимный инструмент)	27
Доступ к учетным данным в процессе «lsass.exe»	16
Обход средств защиты (User Account control)	12
Обход средств защиты (Запуск исполняемого файла через «rundll32.exe» не из системной директории)	6

В результате анализа идентифицированных техник была составлена статистика, приведенная в таблице 5. Таким образом, для исследованной выборки образцов ВПО справедливо заключение, что наиболее обнаруживаемой техникой является «Закрепление в системе». Данная техника встречается более, чем в половине исследованных образцов ВПО. Часто техника «Закрепление в системе» выполняется через различные ключи реестра (например, «Run» или «RunOnce») и папку «Startup Folder». Следующей техникой (для менее, чем трети образцов ВПО), которая детектировалась в ходе экспериментов, является «Обход средств защиты» (запуск через легитимный инструмент). Для реализации этой техники ВПО использовало системные утилиты, например, «rundll32.exe» и «mshta.exe». Третьей достаточно распространенной техникой является «доступ к учетным данным в процессе «lsass.exe». Эта техника возможна благодаря тому, что процесс «lsass.exe», необходимый для аутентификации в ОС Windows, кэширует имена пользователей и хеши их учетных данных. Остальные техники

характерны для оставшейся пятой части исследованной выборки образцов ВПО.

Таким образом, предлагаемая методика позволяет исследователям формировать наборы данных об активности ВПО, используя Sigma-правила для автоматизации разметки событий активности процессов ОС Windows. Проведенные экспериментальные исследования подтверждают эффективность данного подхода, который является отличительной особенностью нашего исследования. Описанная методика позволяет любому исследователю воспроизвести научные результаты авторов настоящей статьи и получить набор данных, необходимый для построения модели машинного обучения, осуществляющей обнаружение ВПО. При этом представленная методика априори предполагает соответствие среды сбора данных реальным условиям работы алгоритма классификации, преобладание привилегий системы сбора данных над привилегиями ВПО и наличие легитимной фоновой активности. Тем самым обеспечивается соответствие формируемого набора данных об активности ВПО критериям качества, введенным в работе [18]. Это является главным преимуществом в сравнении с предшествующими исследованиями. Некоторые предшествующие работы, например, [10, 11] включают в себя создание набора данных с последовательностями API-функций ВПО с помощью Cuckoo Sandbox, при этом разметка по исследуемым файлам производится с помощью сторонних сервисов, логика которых неочевидна. В исследовании [12] приведены этапы работы с данными, но разметка данных не раскрыта подробно.

**7. Заключение.** В данном исследовании затронута актуальная проблема недостаточности полноты и точности обнаружения ВПО существующими методами детектирования ВПО. Для обнаружения ВПО исследователи извлекают статические признаки файлов или динамические признаки процессов ОС Windows (или те, и другие), к которым затем применяют методы детектирования ВПО. В ходе исследования установлено, что значительная часть авторов использует готовые наборы данных с признаками ВПО. Вопросы формирования этих наборов данных в настоящее время раскрыты недостаточно подробно, что ставит под сомнение применимость используемых методов детектирования ВПО. Также важным аспектом формирования набора данных является обработка этих данных и их разметка. Процесс разметки данных является достаточно трудозатратным. В качестве альтернативы в данной работе предложена собственная методика сбора данных об активности ВПО на основе инструмента Sysmon и поведенческих правил Sigma. Использование правил Sigma

позволяет автоматизировать процесс разметки данных. Для проверки применимости данной методики была проведена серия вычислительных экспериментов. Проведенные эксперименты подтвердили возможность использования правил Sigma для разметки вредоносных событий для дальнейшего создания моделей машинного обучения. Направление дальнейшей работы связано с разработкой инструмента валидации правил Sigma и их применения к журналам событий активности ВПО. Другим направлением исследовательской работы является разработка новых правил Sigma, для которых известны существующие техники атаки, но пока отсутствуют готовые правила детектирования.

### Литература

1. Cybercrime Will Cost the World US\$6 Trillion by the End of the Year: Study. URL: <https://cisomag.eccouncil.org/cybercrime-will-cost-the-world-us6-trillion-by-the-end-of-the-year-study/> (дата обращения: 10.11.2023).
2. Ландшафт угроз. URL: <https://encyclopedia.kaspersky.ru/glossary/threat-landscape/> (дата обращения: 08.11.2023).
3. Левшун Д.С., Гайфулина Д.А., Чечулин А.А., Котенко И.В. Проблемные вопросы информационной безопасности киберфизических систем // Информатика и автоматизация. 2020. Т. 19. № 5. С. 1050–1088.
4. ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию // М.: Госстандарт России. 2006.
5. Denning D. An Intrusion-Detection Model // IEEE Transactions on Software Engineering. 1987. no. 2. pp. 222–232.
6. Abaoaja F., Zainal A., Ghaleb F., Al-rimy B.A.S., Eisa T.A.E., Elnour A.A.H. Malware Detection Issues, Challenges, and Future Directions: A Survey // Applied Sciences. 2022. vol. 12. no. 17. pp. 1–29.
7. Herrera-Silva J., Hernandez-Alvarez M. Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms // Sensors. 2023. vol. 23. no. 3. pp. 1–24.
8. Ali R., Ali A., Iqbal F., Hussain M., Ullah F. Deep Learning Methods for Malware and Intrusion Detection: A Systematic Literature Review // Security and Communication Networks. 2022. vol. 2022. pp. 1–31.
9. Gibert D., Mateu C., Planes J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges // Journal of Network and Computer Applications. 2020. vol. 153. pp. 1–22.
10. Kattamuri S., Penmatsa R., Chakravarty S., Madabathula V. Swarm Optimization and Machine Learning Applied to PE Malware Detection towards Cyber Threat Intelligence // Electronics. 2023. vol. 12. no. 2. pp. 1–25.
11. Lu F., Cai Z., Lin Z., Bao Y., Tang M. Research on the Construction of Malware Variant Datasets and Their Detection Method // Applied Sciences. 2022. vol. 12. no. 15. DOI: 10.3390/app12157546.
12. Catak F.O., Yazi A.F., Elezaj O., Ahmed J. Deep learning based Sequential model for malware analysis using Windows exe API Calls // PeerJ Computer Science. 2020. vol. 6. pp. 1–23.
13. Sanchez-Fraga R., Acosta-Bermejo R. Toward a Taxonomy and Multi-label Dataset for Malware Classification // Proceedings of the 10th International Conference in Software Engineering Research and Innovation, CONISOFT. 2022. pp. 150–157.

14. Lee S., Jung W., Lee W., Oh H., Kim E. Android malware dataset construction methodology to minimize bias–variance tradeoff // *ICT Express*. 2022. vol. 8. no. 3. pp. 444–462.
15. Чистяков А., Лобачева Е., Романенко А. Система и способ машинного обучения модели обнаружения вредоносных файлов // Патент RU2673708C1. 2018.
16. Chistyakov A., Lobacheva E., Romanenko A. System and method for generating a convolution function for training a malware detection model // *PATENT USO10922410B2*. 2021.
17. Chhetri S., Hewlett W. Execution behavior analysis text-based ensemble malware detector // *PATENT EP4044054A1*. 2022. pp. 1–26.
18. Rossow C., Dietrich C., Grier C., Paxson V., Pohlmann N., Bos H., Van Steen M. Prudent Practices for Designing Malware Experiments: Status Quo and Outlook // *Proceedings of the 2012 IEEE Symposium on Security and Privacy (S&P 2012)*. 2012. pp. 65–79.
19. Enterprise Matrix. URL: <https://attack.mitre.org/matrices/enterprise/> (дата обращения: 09.11.2023).
20. Sigma. URL: <https://github.com/Neo23x0/sigma> (дата обращения: 05.11.2023).
21. Ica L., Lucian O., Balan T. Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response // *Sensors*. 2023. vol. 23. no. 15. pp. 1–33.
22. Lozano M.A., Llopis I.P., Domingo M.E. Threat Hunting Architecture Using a Machine Learning Approach for Critical Infrastructures Protection // *Big Data and Cognitive Computing*. 2023. vol. 7. no. 2. pp. 1–26.
23. Detecting Process Injection with ETW. URL: <https://web.archive.org/web/20221207000139/https://blog.redbluepurple.io/windows-security-research/kernel-tracing-injection-detection> (дата обращения: 30.10.2023).
24. Schultz M.G., Eskin E., Zadok E., Stolfo S.J. Data mining methods for detection of new malicious executables // *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P 2001)*. 2001. pp. 38–49.
25. Abou-Assaleh T., Cercone N., Keselj V., Sweidan R. N-gram-based detection of new malicious code // *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004)*. 2004. vol. 02. pp. 41–42.
26. Sai M., Tyagi A., Panda K., Kumar S. Machine learning-based malware detection using stacking of opcodes and bytecode sequences // *Proceedings of the 7th International Conference on Parallel, Distributed and Grid Computing (PDGC 2022)*. 2022. pp. 204–209.
27. Hong J., Jeong D., Kim S. Classifying Malicious Documents on the Basis of Plain-Text Features: Problem, Solution, and Experiences // *Applied Sciences*. 2022. vol. 12. no. 8. DOI: 10.3390/app12084088.
28. Tian R., Batten L., Versteeg S. Function length as a tool for malware classification // *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. 2008. pp. 69–76.
29. Dai J., Guha R., Lee J. Efficient Virus Detection Using Dynamic Instruction Sequences // *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS 2008)*. 2008. pp. 69–76.
30. Khalid O., Ullah S., Ahmad T., Saeed S., Alabbad D., Aslam M., Buriro A., Ahmad R. An Insight into the Machine-Learning-Based Fileless Malware Detection // *Sensors*. 2023. vol. 23. no. 2. DOI: [org/10.3390/s23020612](https://doi.org/10.3390/s23020612).
31. Baysa D., Low R.M., Stamp M. Structural entropy and metamorphic malware // *Journal of Computer Virology and Hacking Techniques*. 2013. vol. 9. pp. 179–192.

32. Karim M., Walenstein A., Lakhota A., Parida L. Malware phylogeny generation using permutations of code // *Journal in Computer Virology*. 2005. vol. 1. no. 1-2. pp. 13–23.
33. Botacin M., Galhardo Moia V., Ceschin F., Amaral Henriques M., Gregio A. Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios // *Forensic Science International: Digital Investigation*. 2021. vol. 38. DOI: 10.1016/j.fsidi.2021.301220.
34. Jacob G., Debar H., Filiol E. Behavioral detection of malware: from a survey towards an established taxonomy // *Journal in Computer Virology*. 2008. vol. 4. pp. 251–266.
35. Sgandurra D., Munoz-Gonzalez L., Mohsen R., Lupu E.C. Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection // arXiv. 2016.
36. Abbasi M.S., Al-Sahaf H., Mansoori M., Welch I. Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection // *Applied Soft Computing*. 2022. vol. 121. pp. 1–12.
37. Roesch M. Snort-Lightweight Intrusion Detection for Networks // *Proceedings of the 13th USENIX Conference on System Administration (LISA '99)*. 1999. vol. 99. no. 1. pp. 229–238.
38. YARA. The pattern matching swiss knife for malware researchers (and everyone else). URL: <https://virustotal.github.io/yara/> (дата обращения: 03.11.2023).
39. signature-base. URL: <https://github.com/Neo23x0/signature-base/tree/master/yara> (дата обращения: 03.11.2023).
40. Гайворонская С. Исследование методов обнаружения шеллкодов в высокоскоростных каналах передачи данных // М., дис. канд. ф.-м. наук: 05.13.11: защищена 19.09.2014. 2014. 133 с.
41. Bruschi D., Martignoni L., Monga M. Detecting self-mutating malware using control-flow graph matching // *Detection of Intrusions and Malware & Vulnerability Assessment: Third International Conference*. 2006. pp. 129–143.
42. Naik N., Jenkins P., Savage N., Yang L., Boongoen T., Iam-On N. Fuzzy-import hashing: A static analysis technique for malware detection // *Forensic Science International: Digital Investigation*. 2021. vol. 37. DOI: 10.1016/j.fsidi.2021.301139.
43. Nataraj L., Karthikeyan S., Jacob G., Manjunath B.S. Malware images: visualization and automatic classification // *Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec '11)*. 2011. pp. 1–7.
44. Shabtai A., Moskovitch R., Elovici Y., Glezer C. Detection of malicious code by applying machine learning classifiers on static features, a state-of-the-art survey // *Information Security Technical Report*. 2009. vol. 14. no. 1. pp. 16–29.
45. Sourì A., Hosseini R. A state-of-the-art survey of malware detection approaches using data mining techniques // *Human-centric Computing and Information Sciences*. 2018. vol. 8. no. 1. pp. 1–22. DOI: 10.1186/s13673-018-0125-x.
46. Wagener G., State R., Dulaunoy A. Malware behaviour analysis // *Journal in Computer Virology*. 2008. vol. 4. pp. 279–287.
47. Endpoint Detection and Response (EDR). URL: <https://encyclopedia.kaspersky.com/glossary/edr-endpoint-detection-response/> (дата обращения: 10.11.2023).
48. Шевалье Я., Фендль Ф., Коломеец М.В., Рике Р., Чечулин А.А., Краус К. Обнаружение кибератак в транспортных средствах с использованием характеризующих функций, искусственных нейронных сетей и визуального анализа // *Информатика и автоматизация*. 2021. Т. 20. № 4. С. 845–868.
49. Bostami B., Ahmed M. Deep Learning Meets Malware Detection: An Investigation // *Combating Security Challenges in the Age of Big Data: Powered by State-of-the-Art Artificial Intelligence Techniques*. 2020. pp. 137–155.

50. Зегжда Д.П., Калинин М.О., Крундышев В.М., Лаврова Д.С., Москвин Д.А., Павленко Е.Ю. Применение алгоритмов биоинформатики для обнаружения мутирующих кибератак // Информатика и автоматизация. 2021. Т. 20. № 4. С. 820–844.
51. Jiang J., Zhang F. Detecting Portable Executable Malware by Binary Code Using an Artificial Evolutionary Fuzzy LSTM Immune System // Security and Communication Networks. 2021. vol. 2021. DOI: 10.1155/2021/3578695.
52. Wawryn K., Widulinski P. Detection of anomalies in compiled computer program files inspired by immune mechanisms using a template method // Journal of Computer Virology and Hacking Techniques. 2021. vol. 17. pp. 47–59.
53. Котенко И.В., Саенко И.Б., Лаута О.С., Крибель А.М. Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения // Информатика и автоматизация. 2022. Т. 21. № 6. С. 1328–1358.
54. malware\_images. URL: [http://vision.ece.ucsb.edu/~lakshman/malware\\_images/album/](http://vision.ece.ucsb.edu/~lakshman/malware_images/album/) (дата обращения: 10.11.2023).
55. anubis. URL: <http://anubis.iseclab.org/> (дата обращения: 12.11.2023).
56. Cuckoo Sandbox. URL: <https://cuckoosandbox.org/> (дата обращения: 10.11.2023).
57. ransomwaredataset2016. URL: <https://github.com/rissgroup/ransomwaredataset2016> (дата обращения: 10.11.2023).
58. MalwareDataset. URL: <https://github.com/WindrunnerMax/MalwareDataset> (дата обращения: 13.11.2023).
59. Котенко И., Хмыров С. Анализ моделей и методик, используемых для атрибуции нарушителей кибербезопасности при реализации целевых атак // Вопросы кибербезопасности. 2022. Т. 4. № 50. С. 52–79.
60. The Cyber Kill Chain. URL: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html> (дата обращения: 08.11.2023).
61. The Diamond Model of Intrusion Analysis. URL: <https://www.activereponse.org/wp-content/uploads/2013/07/diamond.pdf> (дата обращения: 06.11.2023).
62. Noor U., Anwar Z., Amjad T., Choo K.-K.R. A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise // Future Generation Computer Systems. 2019. vol. 96. pp. 227–242.
63. Emotet Strikes Again – LNK File Leads to Domain Wide Ransomware. URL: <https://thefirreport.com/2022/11/28/emotet-strikes-again-lnk-file-leads-to-domain-wide-ransomware/> (дата обращения: 03.11.2023).
64. Mandiant. URL: <https://mandiant.com> (дата обращения: 12.11.2023).
65. capa. URL: <https://github.com/mandiant/capa> (дата обращения: 12.11.2023).
66. Levshun D., Kotenko I. A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities // Artificial Intelligence Review. 2023. vol. 56. DOI: 10.1007/s10462-022-10381-4.
67. Check out Check Point’s coverage of the MITRE ATT&CK enterprise matrix. URL: <https://www.checkpoint.com/solutions/mitre-attack/coverage/> (дата обращения: 13.11.2023).
68. Hoglund G., Butler J. Rootkits: Subverting the Windows Kernel: Subverting the Windows Kernel // Addison-Wesley Professional. 2005. 352 p.
69. drakvuf. URL: <https://github.com/tkfenyef/drakvuf> (дата обращения: 09.11.2023).
70. Filter Manager Concepts. URL: <https://learn.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts> (дата обращения: 15.11.2023).
71. About Event Tracing. URL: <https://learn.microsoft.com/en-us/windows/win32/etw/about-event-tracing> (дата обращения: 10.11.2023).
72. Sysmon. URL: <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon/> (дата обращения: 09.11.2023).



73. Security Datasets. URL: <https://github.com/OTRF/Security-Datasets> (дата обращения: 15.11.2023).
74. Sikorski M., Honig A. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software // No Starch Press, 2012. 800 p.
75. community/modules/signatures/ URL: <https://github.com/cuckoosandbox/community/tree/master/modules/signatures/> (дата обращения: 13.11.2023).
76. mongoosecurity. URL: <https://github.com/SDanalytics/mongoosesecurity> (дата обращения: 01.11.2023).
77. What is the ELK Stack? URL: <https://www.elastic.co/what-is/elk-stack> (дата обращения: 15.11.2023).
78. Elastic stack (ELK) on Docker. URL: <https://github.com/deviantony/docker-elk> (дата обращения: 25.11.2023).
79. Elastic Agent 8.1.1. URL: <https://www.elastic.co/downloads/past-releases/elastic-agent-8-1-1> (дата обращения: 03.11.2023).
80. Sigma Command Line Interface. URL: <https://github.com/SigmaHQ/sigma-cli> (дата обращения: 01.08.2023).
81. Types of Malware. URL: <https://www.kaspersky.com/resource-center/threats/malware-classifications> (дата обращения: 09.11.2023).
82. Submission Utility. URL: <https://cuckoo.readthedocs.io/en/latest/usage/submit/?highlight=timeout#submission-utility> (дата обращения: 01.11.2023).

**Смирнов Данил Вадимович** — аспирант, старший преподаватель кафедры, кафедра информационной безопасности киберфизических систем Московского института электроники и математики им. А.Н. Тихонова, Национальный исследовательский университет «Высшая школа экономики». Область научных интересов: исследование поведения вредоносных программ и хакерских атак с использованием методов машинного обучения. Число научных публикаций — 2. DVSmirnov@hse.ru; улица Таллинская, 34, 123458, Москва, Россия; р.т.: +7(495)772-9590.

**Евсютин Олег Олегович** — канд. техн. наук, доцент, заведующий кафедрой, кафедра информационной безопасности киберфизических систем Московского института электроники и математики им. А.Н. Тихонова, Национальный исследовательский университет «Высшая школа экономики»; старший научный сотрудник, лаборатория № 80, ИПУ РАН. Область научных интересов: исследование цифровых водяных знаков и методов их встраивания, цифровая обработка изображений, цифровая стеганография, информационная безопасность. Число научных публикаций — 90. OEvsyutin@hse.ru; улица Таллинская, 34, 123458, Москва, Россия; р.т.: +7(495)772-9590.

D. SMIRNOV, O. EVSUTIN

**METHODOLOGY FOR COLLECTING DATA ON THE ACTIVITY OF MALWARE FOR WINDOWS OS BASED ON MITRE ATT&CK**

*Smirnov D., Evsutin O. Methodology for Collecting Data on the Activity of Malware for Windows OS Based on MITRE ATT&CK.*

**Abstract.** The digitalization of the modern economy has led to the emergence of information technologies in various areas of human activity. In addition to positive effects, this has enhanced the problem of countering cyber threats. The implementation of cyber threats often impacts serious consequences, especially when it comes to critical information infrastructure. Malware is an important part of the modern landscape of cyber threats; the most high-profile cybercrimes of recent years are associated with the use of malware. In this regard, the problem area of countering malware is actively developing, and one of the promising areas of research in this area is the creation of methods for detecting malware based on machine learning. However, the weak point of many well-known studies is the construction of reliable data sets for machine learning models, when the authors do not disclose the features of the formation, preprocessing and labeling of data on malware. This fact compromises the reproducibility a lot of studies. This paper proposes a methodology for collecting data on malware activity based on the MITRE ATT&CK matrix and Sigma rules and designed for Windows OS. The proposed methodology is aimed at improving the quality of datasets containing malware and legitimate processes behavior's features, as well as at reducing the time of data label by an expert method. A software stand was prepared and experiments were carried out for testing the methodology. The results of experiments confirmed applicability of our methodology.

**Keywords:** cybersecurity, malware, MITRE ATT&CK, process activity monitoring, machine learning.

**References**

1. Cybercrime Will Cost the World US\$6 Trillion by the End of the Year: Study. Available at: <https://cisomag.eccouncil.org/cybercrime-will-cost-the-world-us6-trillion-by-the-end-of-the-year-study/> (accessed: 10.11.2023).
2. Threat Landscape. Available at: <https://encyclopedia.kaspersky.ru/glossary/threat-landscape/> (accessed: 08.11.2023).
3. Levshun D., Gaifulina D., Chechulin A., Kotenko I. Problematic issues of information security of cyber-physical systems. SPIIRAS Proceedings. 2020. vol. 19. no. 5. pp. 1050–1088. (In Russ.).
4. GOST R 51275-2006. [Information Security. Informatization object. Factors affecting information], M.: Gosstandart Rossii. 2006. (In Russ.).
5. Denning D. An Intrusion-Detection Model. IEEE Transactions on Software Engineering. 1987. no. 2. pp. 222–232.
6. Aboaaja F., Zainal A., Ghaleb F., Al-rimy B.A.S., Eisa T.A.E., Elnour A.A.H. Malware Detection Issues, Challenges, and Future Directions: A Survey. Applied Sciences. 2022. vol. 12. no. 17. pp. 1–29.
7. Herrera-Silva J., Hernandez-Alvarez M. Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms. Sensors. 2023. vol. 23. no. 3. pp. 1–24.

8. Ali R., Ali A., Iqbal F., Hussain M., Ullah F. Deep Learning Methods for Malware and Intrusion Detection: A Systematic Literature Review. *Security and Communication Networks*. 2022. vol. 2022. pp. 1–31.
9. Gibert D., Mateu C., Planes J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*. 2020. vol. 153. pp. 1–22.
10. Kattamuri S., Penmatsa R., Chakravarty S., Madabathula V. Swarm Optimization and Machine Learning Applied to PE Malware Detection towards Cyber Threat Intelligence. *Electronics*. 2023. vol. 12. no. 2. pp. 1–25.
11. Lu F., Cai Z., Lin Z., Bao Y., Tang M. Research on the Construction of Malware Variant Datasets and Their Detection Method. *Applied Sciences*. 2022. vol. 12. no. 15. DOI: 10.3390/app12157546.
12. Catak F.O., Yazi A.F., Elezaj O., Ahmed J. Deep learning based Sequential model for malware analysis using Windows exe API Calls. *PeerJ Computer Science*. 2020. vol. 6. pp. 1–23.
13. Sanchez-Fraga R., Acosta-Bermejo R. Toward a Taxonomy and Multi-label Dataset for Malware Classification. *Proceedings of the 10th International Conference in Software Engineering Research and Innovation, CONISOFT*. 2022. pp. 150–157.
14. Lee S., Jung W., Lee W., Oh H., Kim E. Android malware dataset construction methodology to minimize bias–variance tradeoff. *ICT Express*. 2022. vol. 8. no. 3. pp. 444–462.
15. Chistyakov A., Lobacheva E., Romanenko A. System and method of machine training model of detecting malicious files. PATENT RU2673708C1. 2018. (In Russ.).
16. Chistyakov A., Lobacheva E., Romanenko A. System and method for generating a convolution function for training a malware detection model. PATENT USO10922410B2. 2021.
17. Chhetri S., Hewlett W. Execution behavior analysis text-based ensemble malware detector. PATENT EP4044054A1. 2022. pp. 1–26.
18. Rossow C., Dietrich C., Grier C., Kreibich C., Paxson V., Pohlmann N., Bos H., Van Steen M. Prudent Practices for Designing Malware Experiments: Status Quo and Outlook. *Proceedings of the 2012 IEEE Symposium on Security and Privacy (S&P 2012)*. 2012. pp. 65–79.
19. Enterprise Matrix. Available at: <https://attack.mitre.org/matrices/enterprise/> (accessed: 09.11.2023).
20. Sigma. Available at: <https://github.com/Neo23x0/sigma> (accessed: 05.11.2023).
21. Ilica L., Lucian O., Balan T. Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response. *Sensors*. 2023. vol. 23. no. 15. pp. 1–33.
22. Lozano M.A., Llopis I.P., Domingo M.E. Threat Hunting Architecture Using a Machine Learning Approach for Critical Infrastructures Protection. *Big Data and Cognitive Computing*. 2023. vol. 7. no. 2. pp. 1–26.
23. Detecting Process Injection with ETW. Available at: <https://web.archive.org/web/20221207000139/https://blog.redbluepurple.io/windows-security-research/kernel-tracing-injection-detection> (accessed: 30.10.2023).
24. Schultz M.G., Eskin E., Zadok E., Stolfo S.J. Data mining methods for detection of new malicious executables. *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P 2001)*. 2001. pp. 38–49.
25. Abou-Assaleh T., Cercone N., Keselj V., Sweidan R. N-gram-based detection of new malicious code. *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004)*. 2004. vol. 02. pp. 41–42.
26. Sai M., Tyagi A., Panda K., Kumar S. Machine learning-based malware detection using stacking of opcodes and bytecode sequences. *Proceedings of the 7th*

- International Conference on Parallel, Distributed and Grid Computing (PDGC 2022). 2022. pp. 204–209.
27. Hong J., Jeong D., Kim S. Classifying Malicious Documents on the Basis of Plain-Text Features: Problem, Solution, and Experiences. *Applied Sciences*. 2022. vol. 12. no. 8. DOI: 10.3390/app12084088.
  28. Tian R., Batten L., Versteeg S. Function length as a tool for malware classification. *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. 2008. pp. 69–76.
  29. Dai J., Guha R., Lee J. Efficient Virus Detection Using Dynamic Instruction Sequences. *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS 2008)*. 2008. pp. 69–76.
  30. Khalid O., Ullah S., Ahmad T., Saeed S., Alabbad D., Aslam M., Buriro A., Ahmad R. An Insight into the Machine-Learning-Based Fileless Malware Detection. *Sensors*. 2023. vol. 23. no. 2. DOI: org/10.3390/s23020612.
  31. Baysa D., Low R.M., Stamp M. Structural entropy and metamorphic malware. *Journal of Computer Virology and Hacking Techniques*. 2013. vol. 9. pp. 179–192.
  32. Karim M., Walenstein A., Lakhota A., Parida L. Malware phylogeny generation using permutations of code. *Journal in Computer Virology*. 2005. vol. 1. no. 1-2. pp. 13–23.
  33. Botacin M., Galhardo Moia V., Ceschin F., Amaral Henriques M., Gregio A. Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios. *Forensic Science International: Digital Investigation*. 2021. vol. 38. DOI: 10.1016/j.fsidi.2021.301220.
  34. Jacob G., Debar H., Filiol E. Behavioral detection of malware: from a survey towards an established taxonomy. *Journal in Computer Virology*. 2008. vol. 4. pp. 251–266.
  35. Sgandurra D., Munoz-Gonzalez L., Mohsen R., Lupu E.C. Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection. arXiv. 2016.
  36. Abbasi M.S., Al-Sahaf H., Mansoori M., Welch I. Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection. *Applied Soft Computing*. 2022. vol. 121. pp. 1–12.
  37. Roesch M. Snort-Lightweight Intrusion Detection for Networks. *Proceedings of the 13th USENIX Conference on System Administration (LISA '99)*. 1999. vol. 99. no. 1. pp. 229–238.
  38. YARA. The pattern matching swiss knife for malware researchers (and everyone else). Available at: <https://virustotal.github.io/yara/> (accessed: 03.11.2023).
  39. signature-base. Available at: <https://github.com/Neo23x0/signature-base/tree/master/yara> (accessed: 03.11.2023).
  40. Gaivoronskaya S. Issledovanie metodov obnaruzheniya shellkodov v vysokoskorostnyh kanalah peredachi dannyh [Investigation of shellcode detection methods in high-speed data transmission channels]. PhD thesis. 2014. 133 p. (In Russ.).
  41. Bruschi D., Martignoni L., Monga M. Detecting self-mutating malware using control-flow graph matching. *Detection of Intrusions and Malware & Vulnerability Assessment: Third International Conference*. 2006. pp. 129–143.
  42. Naik N., Jenkins P., Savage N., Yang L., Boongoen T., Iam-On N. Fuzzy-import hashing: A static analysis technique for malware detection. *Forensic Science International: Digital Investigation*. 2021. vol. 37. DOI: 10.1016/j.fsidi.2021.301139.
  43. Nataraj L., Karthikeyan S., Jacob G., Manjunath B.S. Malware images: visualization and automatic classification. *Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec '11)*. 2011. pp. 1–7.
  44. Shabtai A., Moskovitch R., Elovici Y., Glezer C. Detection of malicious code by applying machine learning classifiers on static features, a state-of-the-art survey. *Information Security Technical Report*. 2009. vol. 14. no. 1. pp. 16–29.

45. Souri A., Hosseini R. A state-of-the-art survey of malware detection approaches using data mining techniques. *Human-centric Computing and Information Sciences*. 2018. vol. 8. no. 1. pp. 1–22. DOI: 10.1186/s13673-018-0125-x.
46. Wagener G., State R., Dulaunoy A. Malware behaviour analysis. *Journal in Computer Virology*. 2008. vol. 4. pp. 279–287.
47. Endpoint Detection and Response (EDR). Available at: <https://encyclopedia.kaspersky.com/glossary/edr-endpoint-detection-response/> (accessed: 10.11.2023).
48. Chevalier Y., Fenzl S., Kolomeets M., Rieke R., Chechulin A., Krauss C. Cyberattack detection in vehicles using characteristic functions, artificial neural networks and visual analysis. *Informatics and Automation*. 2021. vol. 20. no. 4. pp. 845–868.
49. Bostami B., Ahmed M. Deep Learning Meets Malware Detection: An Investigation. *Combating Security Challenges in the Age of Big Data: Powered by State-of-the-Art Artificial Intelligence Techniques*. 2020. pp. 137–155.
50. Zegzhda D., Kalinin M., Krundyshev V., Lavrova D., Moskvina D., Pavlenko E. Application of bioinformatics algorithms for polymorphic cyberattacks detection. *Informatics and Automation*. 2021. vol. 20. no. 4. pp. 820–844. (In Russ.).
51. Jiang J., Zhang F. Detecting Portable Executable Malware by Binary Code Using an Artificial Evolutionary Fuzzy LSTM Immune System. *Security and Communication Networks*. 2021. vol. 2021. DOI: 10.1155/2021/3578695.
52. Wawryn K., Widulinski P. Detection of anomalies in compiled computer program files inspired by immune mechanisms using a template method. *Journal of Computer Virology and Hacking Techniques*. 2021. vol. 17. pp. 47–59.
53. Kotenko I., Saenko I., Lauta O., Kriebel A. Anomaly and cyber attack detection technique based on the integration of fractal analysis and machine learning methods. *Informatics and Automation*. 2022. vol. 21. no 6. pp. 1328–1358. (In Russ.).
54. malware\_images. Available at: [http://vision.ece.ucsb.edu/~lakshman/malware\\_images/album/](http://vision.ece.ucsb.edu/~lakshman/malware_images/album/) (accessed: 10.11.2023).
55. anubis. Available at: <http://anubis.iseclab.org/> (accessed: 12.11.2023).
56. Cuckoo Sandbox. Available at: <https://cuckoosandbox.org/> (accessed: 10.11.2023).
57. ransomwaredataset2016. Available at: <https://github.com/rissgroup/ransomwaredataset2016> (accessed: 10.11.2023).
58. MalwareDataset. Available at: <https://github.com/WindrunnerMax/MalwareDataset> (accessed: 13.11.2023).
59. Kotenko I., Khmyrov S. Analysis of models and techniques used for attribution of cyber security violators in the implementation of targeted attacks. *Voprosy kiberbezopasnosti – Cybersecurity issues*. 2022. vol. 4. no. 50. pp. 52–79.
60. The Cyber Kill Chain. Available at: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html> (accessed: 08.11.2023).
61. The Diamond Model of Intrusion Analysis. Available at: <https://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf> (accessed: 06.11.2023).
62. Noor U., Anwar Z., Amjad T., Choo K.-K.R. A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise. *Future Generation Computer Systems*. 2019. vol. 96. pp. 227–242.
63. Emotet Strikes Again – LNK File Leads to Domain Wide Ransomware. Available at: <https://thedfirreport.com/2022/11/28/emotet-strikes-again-lnk-file-leads-to-domain-wide-ransomware/> (accessed: 03.11.2023).
64. Mandiant. Available at: <https://mandiant.com> (accessed: 12.11.2023).
65. capa. Available at: <https://github.com/mandiant/capa> (accessed: 12.11.2023).

66. Levshun D., Kotenko I. A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities. *Artificial Intelligence Review*. 2023. vol. 56. DOI: 10.1007/s10462-022-10381-4.
67. Check out Check Point's coverage of the MITRE ATT&CK enterprise matrix. Available at: <https://www.checkpoint.com/solutions/mitre-attack/coverage/> (accessed: 13.11.2023).
68. Hoglund G., Butler J. *Rootkits: Subverting the Windows Kernel: Subverting the Windows Kernel*. Addison-Wesley Professional. 2005. 352 p.
69. drakvuf. Available at: <https://github.com/tklengyel/drakvuf> (accessed: 09.11.2023).
70. Filter Manager Concepts. Available at: <https://learn.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts> (accessed: 15.11.2023).
71. About Event Tracing. Available at: <https://learn.microsoft.com/en-us/windows/win32/etw/about-event-tracing> (accessed: 10.11.2023).
72. Sysmon. Available at: <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon/> (accessed: 09.11.2023).
73. Security Datasets. Available at: <https://github.com/OTRF/Security-Datasets> (accessed: 15.11.2023).
74. Sikorski M., Honig A. *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 2012. 800 p.
75. [community/modules/signatures/](https://github.com/cuckoosandbox/community/tree/master/modules/signatures/) Available at: <https://github.com/cuckoosandbox/community/tree/master/modules/signatures/> (accessed: 13.11.2023).
76. [mongoosecurity](https://github.com/SDanilytics/mongoosecurity). Available at: <https://github.com/SDanilytics/mongoosecurity> (accessed: 01.11.2023).
77. What is the ELK Stack? Available at: <https://www.elastic.co/what-is/elk-stack> (accessed: 15.11.2023).
78. Elastic stack (ELK) on Docker. Available at: <https://github.com/deviantony/docker-elk> (accessed: 25.11.2023).
79. Elastic Agent 8.1.1. Available at: <https://www.elastic.co/downloads/past-releases/elastic-agent-8-1-1> (accessed: 03.11.2023).
80. Sigma Command Line Interface. Available at: <https://github.com/SigmaHQ/sigma-cli> (accessed: 01.08.2023).
81. Types of Malware. Available at: <https://www.kaspersky.com/resource-center/threats/malware-classifications> (accessed: 09.11.2023).
82. Submission Utility. Available at: <https://cuckoo.readthedocs.io/en/latest/usage/submit/?highlight=timeout#submission-utility> (accessed: 01.11.2023).

**Smirnov Danil** — Postgraduate student, senior lecturer at the department, Department of information security of cyber-physical systems, Tikhonov Moscow institute of electronics and mathematics, National Research University Higher School of Economics. Research interests: behavior of malware and hacker attacks using machine learning methods. The number of publications — 2. DVSmirnov@hse.ru; 34, Tallinskaya St., 123458, Moscow, Russia; office phone: +7(495)772-9590.

**Evsutin Oleg** — Ph.D., Associate Professor, Head of the department, Department of information security of cyber-physical systems, Tikhonov Moscow institute of electronics and mathematics, National Research University Higher School of Economics; Senior researcher, Laboratory no. 80, ICS RAS. Research interests: digital watermarks and methods of their embedding, digital image processing, digital steganography, information security. The number of publications — 90. OEvsyutin@hse.ru; 34, Tallinskaya St., 123458, Moscow, Russia; office phone: +7(495)772-9590.

П.Д. БОРИСОВ, Ю.В. КОСОЛАПОВ  
**СПОСОБ КОЛИЧЕСТВЕННОГО СРАВНЕНИЯ  
ОБФУСЦИРУЮЩИХ ПРЕОБРАЗОВАНИЙ**

---

*Борисов П.Д., Косолапов Ю.В. Способ количественного сравнения обфусцирующих преобразований.*

**Аннотация.** В работе рассматривается задача количественного сравнения эффективности и стойкости практически применяемых обфусцирующих преобразований программного кода. Предлагается способ нахождения эффективности и стойкости преобразований путем вычисления «понятности» соответственно обфусцированной и деобфусцированной версий программы. В качестве меры понятности программы предлагается использовать похожесть этой программы на аппроксимацию ее «самой понятной» версии. На основе предложенного способа построена модель оценки эффективности и стойкости, основными элементами которой являются: набор исследуемых обфусцирующих преобразований, функция похожести, способ аппроксимации самой понятной версии программы и деобфускатор. Для реализации этой модели 1) выбраны обфусцирующие преобразования, предоставляемые обфускатором Hikari; 2) методами машинного обучения по статическим характеристикам программ из наборов CoreUtils, PolyBench и HashCat построено 8 функций похожести; 3) в качестве аппроксимации самой понятной версии программы выбрана наименьшая по размеру версия программы, найденная среди версий, полученных с помощью опций оптимизации компиляторов GCC, Clang и AOCC; 4) построена и реализована схема деобфускации программ на основе оптимизирующего компилятора из состава LLVM. В работе экспериментально получены результаты оценки эффективности и стойкости для последовательностей преобразований длины один, два и три. Эти результаты показали согласованность с результатами независимых оценок эффективности и стойкости, полученных другими способами. В частности, получено, что наибольшую эффективность и стойкость демонстрируют последовательности преобразований, начинающиеся с преобразований графа потока управления, а наименьшей стойкостью и эффективностью – как правило, последовательности, не содержащие таких преобразований.

**Ключевые слова:** обфускация, исполняемый код, эффективность, стойкость, похожесть.

---

**1. Введение.** Обфусцирующие преобразования обычно применяются для затруднения исследования программ с целью сокрытия алгоритмов и/или данных от аналитика, где под аналитиком может подразумеваться как человек, так и программное средство

автоматического анализа. Такие преобразования применяются как к программам, компилируемым в инструкции целевого процессора (машинный код), так и к интерпретируемым программам (байт-код или промежуточное представление). Способ защиты программ на основе обфускации считается одним из основных в модели MATE (Man At The End), когда нет возможности ограничить аналитика в средствах и методах анализа программного кода [1, 2]. Также обфусцирующие преобразования используются как диверсифицирующие – для защиты от эксплуатации возможных уязвимостей [3].

Исследования в области обфускации ведутся с 70-х годов 20-го века и к настоящему времени выделяются два направления исследований: криптографическое направление (обфускация с доказуемой стойкостью) и направление системного программирования (обфускация с эвристическим обоснованием стойкости) [4]. Криптографами доказано, что не существует запутывающего преобразования, которое бы было стойким в рамках модели «виртуального черного ящика» для всех программ [5]. Не строго говоря, это означает, что для любого запутывающего преобразования найдется программа, по обфусцированному коду которой аналитик может получить больше информации, чем по набору входных и соответствующих выходных данных этой программы. При этом для отдельных классов программ доказуемо стойкая обфускация в модели «виртуального черного ящика» существует [6], а также существуют стойкие запутывающие преобразования для отдельных классов программ в некоторых вариациях этой модели (обзор [4]). Особое место в криптографии занимает неразличимая обфускация, которая позволяет строить стойкие функциональные схемы шифрования [7]. Однако применение неразличимой обфускации на практике затруднено из-за роста размера программы, и пока неясно, насколько такая обфускация затрудняет анализ программ. В целом, имеющиеся результаты в области теоретической (криптографической) обфускации пока далеки от реального применения при защите программ [8]. Напротив, существующие свободно распространяемые и коммерческие средства защиты программ предлагают набор методов обфускации кода, которые получены исследователями, как правило, в области системного программирования. Однако, как замечено в [4], исследования в этой области сравнимы с «состязанием щита и меча»: для используемых на практике способов запутывания кода рано или поздно находятся способы деобфускации (распутывания). Например, обфускация с помощью смешанной булевой и целочисленной арифметики, предложенная в [9], долгое время считалась надежным способом затруднения процесса анализа программ.



Результаты работ [10 – 13] в области упрощения линейных выражений со смешанной булевой и целочисленной арифметикой значительно ослабили позиции этого вида обфускации. В частности, в [13] было показано, что для упрощения могут эффективно применяться методы, используемые при декодировании линейных кодов. Тем не менее, несмотря на отсутствие гарантий стойкости [8], эвристические способы запутывания кода в совокупности с методами противодействия отладке и дизассемблированию могут обеспечить уровень защиты от анализа, достаточный для достижения целей пользователями этих методов [14]: например, анализ программного кода компьютерной игры может потребовать много времени, в течение которого разработчик игры успеет реализовать достаточное для получения прибыли количество лицензий.

В настоящей работе объектом исследования является множество программ  $\mathcal{P}$ , компилируемых в инструкции целевого процессора. Таким образом, в рамках модели МАТЕ, аналитику для исследования доступен бинарный исполняемый файл программы. Для программ из  $\mathcal{P}$  запутывающие преобразования могут применяться на разных уровнях представления: уровне исходного кода, уровне промежуточного представления, уровне машинного кода. Обозначим через

$$\mathcal{O} = \{o_1, \dots, o_m\}$$

набор, далее называемый *базовым*, который может состоять из обфусцирующих преобразований, применяемых на разных уровнях представления. В  $\mathcal{O}$ , например, могут входить такие преобразования, как добавление недостижимого кода, кодирование констант, замена инструкций условного перехода косвенными переходами и т.п. Преобразования, применяемые на одном уровне представления, обычно неприменимы на другом уровне представления, поэтому среди множества всех возможных последовательностей  $\mathcal{O}^*$  выделим множество  $\mathcal{O}_0^*$  всех *допустимых* последовательностей обфусцирующих преобразований. Аналогично множество всех допустимых последовательностей длины  $n$  обозначим  $\mathcal{O}_0^n$ . Очевидно, что  $\mathcal{O}_0^n \subset \mathcal{O}^n$  и  $\mathcal{O}_0^* \subset \mathcal{O}^*$ . Для  $P \in \mathcal{P}$  и  $o \in \mathcal{O}$  через  $o(P)$  обозначим программу, которая получена по  $P$  применением  $o$ .

Для пользователя средств обфускации одной из важных задач является выбор для программы  $P \in \mathcal{P}$  такой последовательности

$$\text{Obf} = (o_{i_1}, o_{i_2}, \dots, o_{i_t}) \in \mathcal{O}_0^*, \quad (1)$$

которая бы при небольшом  $t$  «на приемлемом уровне» затрудняла исследование программы

$$o_{i_t}(\dots(o_{i_1}(P))\dots)$$

аналитиком. Так как для эвристически стойких обфускаций, как отмечалось выше, отсутствует гарантия стойкости, то, с учетом  $|\mathcal{O}_0^*| = \infty$ , под обфусцирующей последовательностью «приемлемого уровня» представляется допустимым понимать ту, которая обладает «наилучшими» характеристиками среди последовательностей заданного набора  $\mathcal{Q} \subset \mathcal{O}_0^*$  конечной мощности. Поэтому для поиска такой последовательности Obf необходим способ *количественного сравнения* результатов применения обфускаций из  $\mathcal{Q}$ .

**2. Обзор известных результатов.** Исследования в области защиты программ представлены достаточно большим числом работ, при этом большая часть работ посвящена вопросам защиты на основе обфускации кода [8]. Считается, что первой работой, систематизировавшей подходы к обфускации исходного кода и оценке качества обфускации, является работа [15]. В [15] *качество* обфусцирующего преобразования определяется как интегральный показатель, учитывающий *эффективность* (potensy в [15]) и *стойкость* (resilience в [15]), где под эффективностью преобразования предложено понимать степень запутанности кода, а под стойкостью – степень запутанности кода после применения средства распутывания (деобфускатора).

В [15] в качестве метрик *эффективности* предлагаются, в частности, метрики из области программной инженерии, такие, как метрики Холстеда. Отметим, что такие метрики, как правило, строятся для уровня исходного кода, и их применение затруднительно для других уровней представления кода. Другой подход рассматривается в [16], где для количественной оценки эффективности предложено использовать степень сжимаемости обфусцированного исходного кода, которая в [16] выбрана в качестве аппроксимации сложности программы по Колмогорову. Результаты экспериментов [16], представленные для преобразований исходного кода программ на языке Java, показали, в частности, что обфусцирующие преобразования над данными оказываются более эффективными, чем преобразования графа потока управления.

При оценке *стойкости* преобразований, если не учитывать оценку, полученную путем опроса пула аналитиков-программистов, сложность заключается в выборе модели деобфускатора. В [17] в качестве такой

модели предложено использовать символьный интерпретатор. Этот подход получил дальнейшее развитие в ряде работ. В частности, в [18] набор  $\mathcal{O}$ , состоящий из девяти преобразований уровня исходного кода, предоставляемых обфускатором Tigress [19] для программ на языке C, поделен на четыре категории, с целью исследования того, как влияет *порядок* применения обфускаций разных категорий (рассматривались все возможные варианты для трех и четырех категорий) на *время* символьного исполнения запутанного тестового кода (в тестовой программе оценивалось время нахождения пароля заданной длины). Таким образом, сравнение стойкости обфусцирующих преобразований в [18] выполняется на основе времени нахождения пароля методами символьной интерпретации. При этом преобразования внутри каждой категории применялись все сразу и в фиксированном порядке, то есть влияние перестановок и комбинаций преобразований внутри каждой категории не исследовалось. Результаты [18] позволяют пользователям Tigress априори выбрать для программы  $P$  подходящую последовательность обфускаций. Однако, проверка того, насколько выбранная последовательность лучше другой для конкретной программы  $P$  затруднена тем, что в [18] для сравнения результатов используется символьная интерпретация, трудно применимая к большим программам [8]. По этой же причине затруднено применение и подходов, предложенных в [20 – 22], где характеристики символьной интерпретации (время интерпретации, объем затраченной оперативной памяти, степень покрытия кода и т.п.) используются для оценки как эффективности, так и стойкости преобразований.

В [8] отмечено, что несмотря на большое количество работ в области обфускации и деобфускации программ, важной и актуальной является задача развития и совершенствования подходов к *количественной* оценке эффективности и стойкости методов защиты. В частности, эти подходы должны учитывать широкое разнообразие техник, применяемых аналитиками при исследовании программ (такие техники включают отладку, дизассемблирование, символьное исполнение, запуск в «песочнице», применение средств тестирования кода и т.п.). Количественная оценка эффективности и стойкости отдельных преобразований и/или их комбинаций позволит подстроить систему защиты под модель аналитика (выбор системы защиты, адекватной возможностям аналитика, является одним из перспективных направлений исследований, по мнению авторов работы [14]).

**3. Постановка задачи.** В настоящей работе ставится задача разработки способа сравнения эффективности и стойкости обфусцирующих преобразований программ на основе *статических* характеристик

бинарного исполняемого файла, то есть таких характеристик, которые могли бы быть получены без запуска или символьной интерпретации программы. Использование характеристик именно исполняемого файла, с одной стороны, соответствует модели МАТЕ, а с другой стороны, исключает зависимость предлагаемого метода от применяемых обфусцирующих преобразований и уровней их применения. Актуальность рассмотрения именно исполняемых файлов также подтверждается наибольшим интересом исследователей к обфускации программ, компилируемым в машинный код, что обосновано большей свободой при построении защиты, чем в случае с интерпретируемыми программами [8]. Отметим, что оценка влияния обфускации на понимание программы *в целом*, а не на понимание отдельной локальной функции, представляется наиболее естественной, так как стойкость к пониманию на локальном уровне не влечет стойкости к пониманию всей программы [8].

**4. «Понятность» программы.** Обфусцирующие преобразования относятся к преобразованиям, сохраняющим семантику программ, к которым принадлежат и оптимизирующие преобразования. Для оптимизирующих преобразований выбор одного из двух  $Opt_1$  и  $Opt_2$  для программы  $P$  может осуществляться, например, так: если размер программы  $Opt_1(P)$  меньше минимума размеров  $P$  и  $Opt_2(P)$ , то наилучшим считается  $Opt_1$ ; аналогично выбор может быть сделан в пользу  $Opt_2$  (в остальных случаях преобразования  $Opt_1$  и  $Opt_2$  не могут рассматриваться как оптимизирующие для  $P$ ). В случае аналогичного выбора одного из двух обфусцирующих преобразований ( $Obf_1$  или  $Obf_2$ ) следует определить *количественную характеристику* программ (бинарных исполняемых файлов), описывающую «понятность» этих программ для аналитика, так как обфусцирующие преобразования направлены на затруднение именно понимания программы (кода и/или данных) [23].

Для введения такой характеристики рассмотрим множество  $\mathcal{P}_P$  программ, полученных из программы  $P$  преобразованиями, сохраняющими семантику. Пусть  $P_0 \in \mathcal{P}$  – самая «понимаемая» версия программы  $P$ . В качестве  $P_0$ , например, можно рассматривать программу наименьшего размера, так как считается, что чем меньше размер программы, тем она «читабельнее» [24]. Другой подход к выбору  $P_0$  может быть основан на сложности символьной интерпретации машинного кода [18, 25]. В таком случае в качестве  $P_0$  может рассматриваться версия с наименьшим временем символьного исполнения или с наименьшим объемом памяти, затраченной при символьном исполнении. В [21] такой подход обоснован тем, что символьное исполнение можно

рассматривать как модель динамического исследования программы аналитиком-человеком. Однако этот подход имеет ограничения: символьное исполнение осуществимо за обозримое время, как правило, только для небольших программ. Понятность программы  $P$  будем рассматривать, как величину похожести программы  $P$  на  $P_0$ . Для этого рассмотрим функцию  $s : \mathcal{P} \times \mathcal{P} \rightarrow [0, 1]$  – функцию похожести программ, где 0 соответствует наименьшей степени похожести, а 1 – наибольшей. В этом случае для фиксированных  $P_0 \in \mathcal{P}_P$  и  $s$  под понятностью программы  $P \in \mathcal{P}_P$  можно подразумевать величину  $s(P, P_0)$ , как степень похожести на самую «понимаемую» версию программы. Однако, как нахождение самой короткой программы, так и нахождение программы с наименьшим временем символьного исполнения, являются вычислительно сложными задачами. Поэтому здесь вместо  $P_0$  предлагается использовать ее аппроксимацию  $A(P_0)$ , найденную по  $P$ . (Аналогичный подход применяется в [16], где при количественной оценке эффективности обфусцирующих преобразований вместо невычислимой колмогоровской сложности программы применяется ее аппроксимация результатом сжатия программы.) Заметим, что аппроксимация  $A(P_0)$  наиболее понятной версии  $P_0$  программы  $P$  – это то, что недоступно аналитику, но если бы было доступно, он смог бы это эффективно понять. В качестве аппроксимации  $A(P_0)$  может быть выбрана, например, наименьшая по размеру версия программы  $P$ , полученная с помощью доступного набора оптимизирующих преобразований. Представляется, что аппроксимацию можно строить также на основе характеристик символьной интерпретации: например, в качестве  $A(P_0)$  может быть выбрана версия программы (например, среди версий, полученных с помощью разных компиляторов и разных опций компиляции) с наименьшим временем символьной интерпретации или наименьшим использованием памяти при символьной интерпретации.

Понятностью  $P \in \mathcal{P}_P$  при фиксированных  $s$  и  $A(P_0) \in \mathcal{P}_P$  назовем величину

$$\text{comp}(P) = s(P, A(P_0)) \in [0, 1]. \quad (2)$$

Таким образом, для зафиксированного алгоритма  $A$  понятность программы  $P$  зависит от выбора функции похожести  $s$ , определенной на парах программ. Известно, что построение функции похожести для программ является нетривиальной задачей и, как правило, зависит от решаемой задачи [26]. Например, если  $s = s^1$ , где  $s^1(P, Q) = 1$  только при совпадении файлов  $P$  и  $Q$ , и равно нулю в остальных случаях, то

понятность большинства программ будет равна нулю. Другим крайним случаем функции похожести является функция  $s = s^2$ , такая, что  $s^2(P, Q) = 1$ , если  $P$  и  $Q$  на одинаковых входных данных возвращают одинаковые результаты, и равно нулю в остальных случаях. В этом случае понятность большинства программ (всех программ, если  $A$  сохраняет функциональность) будет максимальна. Однако функции  $s^1$  и  $s^2$ , как будет следовать из определений эффективности и стойкости обфусцирующих преобразований (определения (3)) и (4) в разделе 5), малоприспособны для решения поставленной в разделе 3 задачи: при использовании  $s^1$  все обфусцирующие преобразования в равной степени будут эффективными и стойкими, а при использовании  $s^2$  ни одно из обфусцирующих преобразований не будет ни эффективным, ни стойким. Однако практические исследования показывают (например, [27, 28]), что разные запутывающие преобразования могут по-разному влиять на время, затрачиваемое аналитиком для понимания обфусцированной программы. Поэтому для решаемой задачи актуально построение таких функций похожести, значения которых бы коррелировали с результатами практических исследований эффективности и стойкости обфусцирующих преобразований. При этом для рассматриваемой задачи важную роль играет то, как такие функции могут оценить степень похожести программ, реализующих один и тот же алгоритм, и в меньшей степени важно то, как эти функции оценивают степень похожести разных по функционалу программ. В [29] авторами предпринята попытка построения такой функции на основе методов машинного обучения с использованием множества известных функций похожести, определенных на парах бинарных файлов. Эта функция используется в настоящей работе как конкретная реализация абстрактной функции похожести, используемой в определении понятности программы  $\text{comp}(P)$  (более подробное описание функции приведено в разделе 5.2). Заметим, что значение  $\text{comp}(P)$ , когда к  $P$  не применяются преобразования, как представляется, может использоваться, например, в программной инженерии для оценки качества программного кода: чем больше  $\text{comp}(P)$ , тем качество выше, и наоборот.

### **5. Модель оценки эффективности и стойкости обфускации.**

С помощью характеристики (2) определим эффективность  $e$  обфусцирующего преобразования  $\text{Obf} \in \mathcal{O}_0^*$ , примененного к программе  $P$ , а также стойкость  $r$  этого преобразования по отношению к деобфускатору  $D$  следующим образом:

$$e(\text{Obf}, P) = 1 - \text{comp}(\text{Obf}(P)) = 1 - s(\text{Obf}(P), A(P_0)), \quad (3)$$

$$r(D, \text{Obf}, P) = 1 - \text{comp}(D(\text{Obf}(P))) = 1 - s(D(\text{Obf}(P)), A(P_0)). \quad (4)$$

В рамках определений (3) и (4), задача выбора наиболее эффективного обфусцирующего преобразования для  $P$  решается так:  $\text{Obf}_1$  эффективнее  $\text{Obf}_2$ , если

$$e(\text{Obf}_1, P) > \max\{1 - \text{comp}(P), e(\text{Obf}_2, P)\}.$$

Задача выбора среди этих же преобразований наиболее стойкого по отношению к деобфускатору  $D$  решается так:  $\text{Obf}_1$  является  $D$ -устойчивее  $\text{Obf}_2$ , если

$$r(D, \text{Obf}_1, P) > r(D, \text{Obf}_2, P).$$

Набор

$$\mathcal{M} = (\mathcal{O}, \mathcal{Q} \subset \mathcal{O}_0^*, s, D, A) \quad (5)$$

с определенными в соответствии с (3) и (4) функциями  $e$  и  $r$  назовем *моделью оценки эффективности и стойкости обфусцирующих преобразований*. Для реализации этой модели необходимо зафиксировать множество  $\mathcal{O}$ , выбрать функцию похожести  $s$  исполняемых файлов программ, деобфускатор  $D$  и способ аппроксимирования  $A$  самой понятной программы  $P_0 \in \mathcal{P}_P$  для каждого  $P$ .

**5.1. Выбор множеств  $\mathcal{O}$  и  $\mathcal{Q}$ .** В качестве множества  $\mathcal{O}$  в настоящей работе выбрано множество преобразований, предоставляемых свободно распространяемым обфускатором Nikari [30], основанным на обфускаторе Obfuscator-LLVM [31]. Обозначения и соответствующие описания выбранных преобразований представлены в таблице 1. В последнем столбце указан тип преобразования, который выбран в соответствии с определениями типов преобразований, введенных в [18] для обфускатора Tigress:  $A$  – абстрактные преобразования,  $C$  – преобразования графа потока управления,  $D$  – кодирование данных и арифметических операций. Для  $o \in \mathcal{O}$  тип этого преобразования будем обозначать  $\theta(o)$ , а тип последовательности  $\text{Obf}$  вида (1) естественно определить так:  $\theta(\text{Obf}) = (\theta(o_{i_1}), \dots, \theta(o_{i_t}))$ . Например, тип последовательности  $\text{Obf} = (\text{ind}, \text{fcw}, \text{sbb})$  будет  $(C, C, A)$ . Для простоты записи тип последовательности будем указывать без скобок и запятых.

То есть в рассмотренном примере тип будет ССА. Множество  $\{C,A,D\}$  обозначим  $\theta(\mathcal{O})$ , как множество типов базовых преобразований из  $\mathcal{O}$ .

Таблица 1. Набор  $\mathcal{O}$  преобразований, предоставляемых компилятором Nikarі для программ на языке  $C$

	$o$	Описание преобразования	Тип
$\mathcal{O}$	bcf	встраивание непрозрачных предикатов	C
	cff	сглаживания графа потока управления	C
	enc	кодирование статических строк	D
	fcw	создание фиктивных функций-прокси	A
	ind	замена инструкций ветвления косвенными переходами	C
	sbb	разбиение базовых блоков	A
	sub	замена инструкций эквивалентными	D
	all	применение всех обфусцирующих преобразований	–

Отметим, что все выбранные преобразования из  $\mathcal{O}$  применяются на уровне промежуточного представления (на уровне IR-кода), однако порядок применения преобразований не является произвольным, поэтому  $\mathcal{O}_0^* \neq \mathcal{O}^*$ . При нахождении наиболее эффективного и/или стойкого обфусцирующего преобразования  $Obf \in \mathcal{Q}$  естественно предполагать, что выбор осуществляется среди последовательностей вида (1) одной и той же длины  $t$ . Однако перебор всех  $|\mathcal{O}|^t$  последовательностей длины  $t$  уже при небольших значениях  $t$  может потребовать значительных вычислительных ресурсов. В настоящей работе полный перебор возможных последовательностей осуществляется для  $t = 1, 2, 3$ . Дополнительно рассматривается последовательность all длины 7, в которой применяются все преобразования из набора  $\mathcal{O}$ . Преобразование all добавлено в  $\mathcal{Q}$  как референсное, так как априорно ожидается, что это преобразование является наиболее эффективным и стойким, поскольку задействованы все преобразования из  $\mathcal{O}$  (хотя и в фиксированном порядке).

**5.2. Выбор функции похожести.** Разнообразие подходов к сравнению бинарных данных велико (обзор [32]). Для реализации модели  $\mathcal{M}$  может подойти практически любая функция похожести, определенная на парах бинарных файлов. Эти функции могут как учитывать то, что входными данными являются исполняемые файлы (учитывается структура и характер данных в файле – *специализированные* характеристики), так и не учитывать (учитываются *универсальные* для любых данных характеристики). В первом случае примерами могут быть функции сравнения графов потока управления, полученных с помощью дизассемблера [34], или функции на основе характеристик символического исполнения кода [18, 21, 22]. Во втором случае примерами функций



похожести являются функции на основе нечетких хеш-функций [33], а также функции на основе алгоритмов сжатия [16]. В настоящей работе используется разработанный авторами в [29] подход *статического* вычисления похожести файлов с использованием методов машинного обучения на основе функций похожести, вычисляемых от статических универсальных и специализированных характеристик бинарных исполняемых файлов. Выбор в пользу статических характеристик сделан по причине применимости этого подхода к программам любого размера и сложности. Обозначение выбранных функций и их краткое описание дано в таблице 2.

Таблица 2. Статические характеристики похожести программ  $P$  и  $Q$

$f$	Значение характеристики	$Accuracy(s_f)$	$Impact(f)$
sdhash	результат сравнения хеш-значений функции sdhash для $P$ и $Q$	0.50	0.01
tlsh	результат сравнения хеш-значений функции tlsh с учетом длины $P$ и $Q$	0.68	0.06
tlshxlen	результат сравнения хеш-значений функции tlsh без учета длины $P$ и $Q$	0.53	0.02
lzma	коэффициент относительного сжатия алгоритмом lzma программ $P$ и $Q$	0.66	0.03
bz2	коэффициент относительного сжатия алгоритмом bz2 программ $P$ и $Q$	0.65	0.02
deflate	коэффициент относительного сжатия алгоритмом deflate программ $P$ и $Q$	0.68	0.03
myersdiff	расстояние Майерса для $P$ и $Q$	0.65	0.05
levenshteindiff	расстояние Левенштейна для $P$ и $Q$	0.67	0.06
idadiff	результат сравнения утилитой BinDiff графов потока управления $P$ и $Q$ , полученных дизассемблером IDA Pro	0.66	0.12
idadiffc	произведение idadiff на степень уверенности, возвращаемую BinDiff	0.69	0.14
ghydradiff	результат сравнения утилитой BinDiff графов потока управления $P$ и $Q$ , полученных дизассемблером Ghyptra	0.64	0.11
ghydradiffc	произведение ghydradiff на степень уверенности, возвращаемую BinDiff	0.68	0.14
ninjadiff	результат сравнения утилитой BinDiff графов потока управления $P$ и $Q$ , полученных дизассемблером Ninja	0.56	0.09
ninjadiffc	произведение ninjadiff на степень уверенности, возвращаемую BinDiff	0.66	0.12

Более подробное описание способов вычисления значений этих функций можно найти в [29]. Отметим, что здесь, в отличие [29], используются четырнадцать функций, а не пятнадцать, так как

результаты [29] показывают, что вклад значений хеш-функции `ssdeep` практически не влияет на точность классификатора. Также отметим, что в [29] для функции похожести, как некоторого аналога «расстояния» между программами, предельным значениям 0 и 1 поставлено соответствие, обратное принятому в настоящей работе: нулю соответствует совпадение программ, а единице – различие. Здесь же для удобства определения понятности программы применена инверсия предельных значений: чем более программа  $P$  похожа на  $A(P_0)$ , тем более она понятна. В таблице 2 универсальные характеристики (первые восемь) отделены от специализированных двойной линией. В качестве методов машинного обучения использованы классификаторы, описанные в таблице 3.

Таблица 3. Классификаторы, лежащие в основе соответствующих функций похожести

$c$	Описание классификатора	$Accuracy(s_c)$
RFC	метод «случайный лес»	0,999
KNC	метод $K$ ближайших соседей	0,996
SVC	метод опорных векторов	0,985
DTC	деревья решений	0,974
LGC	линейная регрессия	0,966
LDAC	линейный дискриминантный анализ	0,962
GNBC	наивная байесовская классификация	0,716
NNC	нейронная сеть с двумя скрытыми слоями по 10 и 100 нейронов соответственно	0,962

Далее функцию похожести, построенную на основе классификатора  $c$  из таблицы 3 будем обозначать  $s_c$ . В случае, когда классификатор строится с помощью только одной характеристики  $f$  из таблицы 2, соответствующую функцию похожести будем обозначать  $s_f$ .

**5.3. Выбор деобфускатора.** В рамках модели МАТЕ, когда аналитику доступен бинарный исполняемый файл, деобфускатор  $D$ , по отношению к которому оценивается стойкость обфусцирующих преобразований, должен принимать на вход машинный код. Так как оценка стойкости в соответствии с (4) предполагает сравнение деобфусцированной версии  $D(\text{Obf}(P))$  с аппроксимацией  $A(P_0)$ , то выходом деобфускатора  $D$  должен быть также бинарный исполняемый файл. Такую возможность предоставляет средство SATURN [35], авторы которого позиционируют его как «state-of-the-art» деобфускатор бинарных исполняемых файлов. Идея, лежащая в основе этого деобфускатора заключается в получении с помощью дизассемблера IDA Pro представления программы в виде графа (включает функции, их

программный код, глобальные переменные/данные и т.д.), трансляции полученного представления в промежуточное представление LLVM с помощью библиотеки Remill из состава McSema [36], применении деобфусцирующих преобразований и компиляции полученного кода в бинарный исполняемый файл. При этом в основе операций деобфускации лежат *оптимизирующие* преобразования. Как отмечено в [25], «трансляция машинного кода» до IR-кода с помощью Remill, его оптимизация с последующей трансляцией обратно в машинный код приводит к тому, что полученная программа представляет собой *интерпретацию* машинных команд исходной программы с помощью последовательностей также машинных команд. Такая трансформация приводит к увеличению программного кода и, как следствие, замедлению работы программы. В [35] применяются оптимизирующие техники для снижения эффекта «интерпретации», однако трудность независимой оценки работы деобфускатора SATURN заключается в отсутствии опубликованного кода этого средства.

Отметим, что идея использования оптимизирующих преобразований, как упрощающих или деобфусцирующих, выдвигалась, помимо работы [35], также в других работах, в частности, в [37, 38, 25]. В настоящее время перечень средств, оптимизирующих непосредственно машинный код, нет так много. Одним из немногих таких оптимизаторов является средство BOLT [39] и его усовершенствованная версия VESPA [40]. Оптимизирующие (упрощающие) преобразования этих средств заключаются в основном в упорядочивании базовых блоков и функций программы таким образом, чтобы вызываемые «по соседству» функции находились в одной странице кода. Это снижает время доступа к инструкциям, так как в кэш процессора код загружается страницами. Однако других упрощений кода (например, удаление недостижимого и мусорного кода, упрощение арифметических выражений и т.п.) не производится, поэтому оптимизаторы BOLT и VESPA вряд ли могут претендовать на роль деобфускатора – средства, упрощающего понимание программы.

В настоящей работе на основе LLVM построена схема получения деобфусцированного кода, показанная на рисунке 1. Идея, лежащая в основе этой схемы состоит в применении оптимизирующих преобразований к обфусцированному IR-коду. Такой подход, с одной стороны, исключает интерпретацию машинных инструкций машинными инструкциями, как это происходит при «трансляции машинного кода» в IR-код, с другой – требует только указания дополнительных опций компилятору LLVM.



Рис. 1. Модель генерации программ  $P$ ,  $Obf(P)$ , и  $D(Obf(P))$  с помощью LLVM

**5.4. Выбор подхода аппроксимирования.** В разделе 4 отмечено, что аппроксимацией  $A(P_0)$  самой понятной версии  $P_0$  для программы  $P$  можно считать, например, 1) аппроксимацию наименьшей по размеру версии, или 2) аппроксимацию версии с наименьшим временем символического исполнения. Повторим, что эвристически первый подход может быть обоснован тем, что короткий код считается наиболее читабельным [24]; второй же подход может быть обоснован наличием опубликованных работ, где обфусцирующие преобразования оцениваются при помощи характеристик символического исполнения [18, 25]. Однако недостатком второго подхода является то, что при символическом исполнении даже небольших программ с малым числом входных параметров, как замечено в [21], затрачиваются значительные вычислительные ресурсы (процессорное время, объем оперативной памяти). Это может препятствовать нахождению  $A(P_0)$  для больших программ, и программ, имеющих большое число входных параметров. Нахождение же  $A(P_0)$ , как аппроксимации самой короткой версии программы  $P$ , можно эффективно реализовать, например, следующим образом: 1) скомпилировать  $P$  с помощью *разных компиляторов*, для каждого из которых использовать *доступное множество опций оптимизации*; 2) среди полученных версий программы  $P$  выбрать программу с наименьшим размером; это и будет  $A(P_0)$ . В настоящей работе использованы компиляторы и опции компиляции, представленные

в таблице (4). Таким образом, для программы  $P$  аппроксимация  $A(P_0)$  находится среди 45 версии этой программы.

Таблица 4. Используемые компиляторы и опции компиляции: 9 компиляторов, для каждого из которых применяются 5 опций компиляции (опций оптимизации)

Компилятор	Версии	Опции
GCC [42]	7.5.0, 8.4.0, 9.4.0, 10.3.0	O0, O1
Clang [43]	7.0.1, 8.0.1, 9.0.1, 10.0.0	O2, O3,
АОСС [44]	3.0.0	Os

**6. Экспериментальные исследования.** Целью экспериментов является исследование применимости на практике предлагаемого метода оценки эффективности и стойкости обфусцирующих преобразований. Другими словами, исследуется корреляция между значением  $s(\text{Obf}(P), A(P_0))$  и «реальной» похожестью  $\text{Obf}(P)$  на  $A(P_0)$  (в случае оценки эффективности), а также корреляция между  $s(D(\text{Obf}(P)), A(P_0))$  и «реальной» сложностью деобфускации  $\text{Obf}(P)$ , выражаемой как похожесть  $D(\text{Obf}(P))$  на  $A(P_0)$  (в случае оценки стойкости к деобфускатору). Трудность такого исследования заключается в нехватке независимых результатов, которые могли бы быть референсными («реальными») при оценке эффективности и стойкости. Причина этой нехватки, вероятно, заключается в том, что реальная эффективность и стойкость может быть оценена только человеком, а такие исследования требуют больших ресурсов, в том числе, временных. В этом направлении стоит отметить работы [27, 28, 41], где оценка эффективности и стойкости обфусцирующих преобразований выполнена с привлечением аналитиков. Отметим, во всех трех работах исследовалась обфускация исходного кода. Целью работ [27, 41] было исследование влияния обфускации в целом на сложность понимания и изменения исходного кода нескольких программ (на языках Java и C/C++), а не влияние отдельных преобразований. Учитывая то, что в настоящей работе аналитику доступен машинный код, а не исходный, а так же то, что целью работы является механизм сравнения отдельных преобразований, результаты работ [27, 41] не могут использоваться как референсные. Результаты работы [28] с одной стороны могут быть использованы, так как исследуется влияние двух обфусцирующих преобразований: переименовывание идентификаторов и вставка непрозрачных предикатов, с другой – использование может быть частичным, так как обфусцирующие преобразования исходного кода могут быть нивелированы на уровне машинного кода (например, переименовывание идентификаторов в

исходном коде может быть частично нивелировано опциями компиляции в машинном коде).

Учитывая растущее разнообразие обфусцирующих преобразований и подходов к деобфускации, вряд ли можно ожидать появление масштабных исследований эффективности и стойкости с помощью человека. В то же время вполне ожидаемыми являются исследования, в которых действия человека моделируются автоматическими средствами. Одной из таких моделей является символьное исполнение [17, 25]. Поэтому в настоящей работе при оценке эффективности и стойкости обфусцирующих преобразований в качестве референсных в дополнение к [28] используются результаты из работ [18, 20, 21].

**6.1. Параметры модели.** В работе исследуется модель (5) при  $Q = \mathcal{O}_0^1 \cup \mathcal{O}_0^2 \cup \mathcal{O}_0^3 \cup \{\text{all}\}$ , где  $\mathcal{O}$  – набор преобразований из таблицы 1. Отметим, что  $|Q| = 64$ . Данные (программы), используемые в экспериментальном исследовании модели (5), условно можно разделить на две части: 1) данные, необходимые для обучения бинарных классификаторов, на основе которых строятся функции похожести  $s = s_c$  и  $s = s_f$ ; 2) данные, необходимые для оценки стойкости и эффективности обфусцирующих преобразований из множества  $Q$ . Стоит отметить, что построение набора программ, который можно было бы использовать для независимой оценки средств защиты, является отдельной задачей [8]. В настоящей работе делается шаг в направлении построения набора программ для оценки обфусцирующих преобразований.

Функции похожести  $s_c$  и  $s_f$ , где  $c$  – классификатор из таблицы 3,  $f$  – признак из таблицы 2, построены (обучены) на основе 164 программ, указанных в таблице 5.

Таблица 5. Набор программ для построения функций похожести

Набор	Количество
CoreUtils [45]	106
PolyBench [46]	30
HashCat [47]	28

Отметим, что набор CoreUtils часто применяется для оценки обфусцирующих преобразований, однако в [8] отмечено, что только утилиты операционной системы вряд ли могут являться репрезентативным набором программ, результаты для которого могут быть перенесена на другие типы программы. Поэтому в настоящей работе кроме CoreUtils используются два других набора: набор программ PolyBench, реализующий алгоритмы компьютерной алгебры, и набор HashCat, реализующий алгоритмы криптографических хеш-функций.

С помощью 9-ти компиляторов и 5-ти опций компиляции, указанных в таблице 4, построен набор из  $9 \cdot 5 \cdot 164 = 2880$  программ. По этому набору построено множество пар похожих программ  $\mathcal{S}$ , состоящее из  $\binom{9 \cdot 5}{2} \cdot 164 = 162360$  пар, а также множество пар разных программ  $\mathcal{D}$ , состоящее из  $\binom{164}{2} \cdot (9 \cdot 5) = 601470$  пар (рассматривались только сравнения разных программ для одинаковых компиляторов и опций оптимизации, поэтому множитель  $9 \cdot 5$ , а не  $(9 \cdot 5)^2$ ).

На основе  $\mathcal{S}$  и  $\mathcal{D}$  построены набор  $\mathcal{F}$  векторов длины 15, в каждый из которых входят 14 значений признаков из таблицы 2 и метка похожести  $M$  (для пар из  $\mathcal{S}$  метка  $M$  имеет значение 1, а для пар из  $\mathcal{D}$  – значение 0). В набор  $\mathcal{F}$  входят векторы, соответствующие всем парам из  $\mathcal{S}$ , а также набор из случайно выбранных  $|\mathcal{S}|$  пар из  $\mathcal{D}$ . Кроме этого в набор  $\mathcal{F}$  включены два вектора: вектор с меткой 0, состоящий из нулей, а также вектор с меткой 1, состоящий из единиц. Эти векторы включены как крайние случаи: вектор из нулей соответствует разным программам, а вектор из единиц похожим программам. Таким образом, набор  $\mathcal{F}$  состоит из  $324722 = 2|\mathcal{S}| + 2$  векторов длины 15. Напомним, что для функций  $s_c$  векторы признаков имеют размерность 15 (14 признаков из таблицы 2 и бинарная метка похожести  $M$ ), в то время как вектор признаков для функций  $s_f$  имеет размерность два (один признак  $f$  из таблицы 2 и метка  $M$ ). Как показали результаты обучения на наборе  $\mathcal{F}$ , функции  $s_c$  обладают большей точностью, чем любая из функций  $s_f$  (столбцы  $Accuracy(s_c)$  и  $Accuracy(s_f)$  в таблицах 3 и 2 соответственно). В таблице 2 в столбце  $Impact(f)$  приведена важность характеристики  $f$ , рассчитанная с помощью оценки взаимной информации между характеристикой  $f$  и меткой похожести  $M$ . Видно, что наибольшей важностью обладают характеристики, полученные с помощью дизассемблирования программ (специализированные характеристики).

Для исследования эффективности и стойкости обфусцирующих преобразований используются два множества: множество  $\mathcal{P}_1^{test}$ , составленное из программ, входящих в наборы из таблицы 5, а также множество  $\mathcal{P}_2^{test}$ , составленное из программ, не входящих в указанные наборы. В настоящей работе множество  $\mathcal{P}_2^{test}$  состоит из 20 программ, которые являются программами с открытым исходным кодом из набора [48]. Множество  $\mathcal{P}_1^{test}$  строится путем случайного выбора также 20-ти программ из 58-ми программ, входящих в наборы PolyBench и NashCat из таблицы 5. По набору  $\mathcal{P}_i^{test}$ ,  $i = 1, 2$ , строится набор  $\mathcal{F}_i$  векторов признаков, соответствующих парам  $(Obf(P), A(P_0))$ , где  $P \in \mathcal{P}_i^{test}$ ,  $Obf \in \mathcal{Q}$ . Таким образом,  $|\mathcal{F}_i| = 1280$ ,  $i = 1, 2$ . Аппроксимация  $A(P_0)$  находится в соответствии с описанным в разделе

5.4 подходом: как наименьшая по размеру версия программы  $P$  среди 45 версий. Отметим, что, как показали эксперименты, чаще  $A(P_0)$  соответствует версии  $P$ , полученной с помощью компиляторов АОСС и GCC с опцией Os: 98% аппроксимаций получены с помощью компилятора АОСС, а остальные – с помощью различных версий компилятора GCC.

**6.2. Исследование эффективности обфусцирующих преобразований.** В таблице 6 для каждого из  $\text{Obf} \in \mathcal{Q}$  (столбцы с именем  $\text{Obf}$ ) в столбцах с именами  $e_1$  и  $e_2$  приведено среднее значение эффективности, вычисленное по формуле (3) на основе функций похожести  $s_c$  для всех классификаторов  $c$  из таблицы 3. Первый и третий столбец таблицы содержит преобразования, упорядоченные по убыванию значения эффективности  $e_1$  (от более эффективных к менее эффективным, или, что то же самое, от менее похожих на аппроксимацию к более похожим), полученного на множестве  $\mathcal{F}_1$ , а второй и четвертый – преобразования, упорядоченные по убыванию значения  $e_2$ , полученного на множестве  $\mathcal{F}_2$ . Далее, под обозначением  $e_i(\text{Obf})$  будем понимать значение из столбца с именем  $e_i$  таблицы 6, соответствующее преобразованию  $\text{Obf} \in \mathcal{Q}$  и полученное на множестве  $\mathcal{F}_i, i = 1, 2$ .

На основе данных из таблицы 6 вычислен рейтинг эффективности отдельных преобразований, отдельных пар преобразований и отдельных троек преобразований, входящих в обфусцирующие последовательности. Рейтинг эффективности преобразований рассчитан двумя способами: с учетом порядка следования базовых преобразований и без учета этого порядка. В первом случае набор преобразований  $n = (X_1, \dots, X_v) \in \mathcal{O}^v$  считается входящим в последовательность  $\text{Obf} = (o_{i_1}, \dots, o_{i_t})$ , если существует  $j, 1 \leq j \leq t - v$ , что  $o_{i_j} = X_1, o_{i_{j+1}} = X_2, \dots, o_{i_{j+v}} = X_v$ . Во втором случае этот набор считается входящим в последовательность  $\text{Obf}$ , если существуют такие  $1 \leq j_1 < \dots < j_v \leq t$ , что  $o_{i_{j_1}} = X_1, o_{i_{j_2}} = X_2, \dots, o_{i_{j_v}} = X_v$ . Вхождение набора  $n$  в  $\text{Obf}$ , учитывающее порядок, будем обозначать  $n \in_+ \text{Obf}$ , а вхождение без учета порядка – соответственно  $n \in_- \text{Obf}$ . Для набора  $n$  его рейтинг эффективности для заданных  $\Delta \in \{+, -\}$  и  $i = 1, 2$  определим формулой:

$$\rho_{\Delta, i}(n) = \frac{\sum_{\text{Obf} \in Q_{\Delta}(n)} e_i(\text{Obf})}{|Q_{\Delta}(n)|},$$

где  $Q_{\Delta}(n)$  – множество обфусцирующих последовательностей из  $\mathcal{Q} \setminus \{\text{all}\}$ , содержащих с учетом порядка или без учета (в зависимости от  $\Delta$ ) преобразование  $n$ :



$$Q_{\Delta}(n) = \{\text{Obf} \in Q \setminus \{\text{all}\} : n \in_{\Delta} \text{Obf}\}. \quad (6)$$

В таблице 7 приведен вычисленный рейтинг эффективности одиночных преобразований; для сравнения в таблицу добавлено соответствующее значение для преобразования all из таблицы 6.

Таблица 6. Упорядоченные по убыванию значений эффективности  $e_1$  и  $e_2$  обфусцирующие преобразования длины один, два и три

$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
Obf	$e_1$	Obf	$e_2$	Obf	$e_1$	Obf	$e_2$
ind,fcw,sbb	0,85	all	0,68	bcf,sbb	0,335	ind,cff	0,364
bcf,ind,fcw	0,802	bcf,ind,fcw	0,601	bcf,cff,sub	0,324	ind,cff,sub	0,364
all	0,797	bcf,fcw,sbb	0,583	enc,bcf,sub	0,313	bcf,fcw,sub	0,358
enc,ind,fcw	0,735	bcf,ind,sbb	0,574	enc,bcf,cff	0,308	bcf	0,355
bcf,ind,sub	0,732	enc,bcf,ind	0,544	enc,bcf	0,307	enc,fcw,sbb	0,335
ind,fcw	0,65	ind,fcw,sbb	0,531	enc,fcw	0,304	bcf,sub	0,317
enc,ind,sbb	0,626	bcf,cff,sbb	0,528	enc,fcw,sub	0,291	enc,sbb	0,31
bcf,ind	0,623	enc,bcf,sbb	0,508	bcf,sub	0,29	enc,fcw,sub	0,309
ind,fcw,cff	0,617	bcf,sbb	0,501	fcw	0,288	enc,cff,sbb	0,306
bcf,ind,cff	0,611	bcf,ind,cff	0,486	fcw,cff,sub	0,277	enc,sub,sbb	0,304
ind,sub,sbb	0,611	bcf,ind	0,477	enc,fcw,cff	0,272	enc,fcw,cff	0,303
ind,sbb	0,607	ind,sub,sbb	0,476	bcf	0,271	enc,sub	0,301
ind,fcw,sub	0,6	ind,cff,sbb	0,473	fcw,sub	0,269	enc	0,298
ind,cff,sbb	0,59	ind,sbb	0,473	fcw,cff,sbb	0,268	enc,fcw	0,295
bcf,fcw	0,576	bcf,sub,sbb	0,472	fcw,cff	0,267	fcw,sbb	0,291
enc,bcf,ind	0,564	bcf,fcw	0,472	fcw,sub,sbb	0,259	enc,cff	0,287
enc,bcf,fcw	0,563	enc,ind,sbb	0,471	fcw,sbb	0,25	fcw,sub,sbb	0,285
bcf,ind,sbb	0,559	bcf,fcw,cff	0,465	bcf,cff	0,24	fcw,cff,sbb	0,281
ind,sub	0,557	enc,ind,fcw	0,462	enc,sbb	0,178	fcw,cff,sub	0,279
ind	0,549	bcf,ind,sub	0,453	enc,cff	0,176	fcw	0,278
ind,cff,sub	0,541	enc,bcf,fcw	0,445	enc,cff,sbb	0,175	enc,cff,sub	0,277
enc,ind	0,538	ind,fcw,sub	0,434	enc	0,175	fcw,cff	0,275
enc,ind,cff	0,538	ind,fcw,cff	0,43	enc,cff,sub	0,174	bcf,cff,sub	0,273
bcf,fcw,cff	0,534	enc,ind	0,422	enc,sub,sbb	0,169	fcw,sub	0,272
ind,cff	0,526	ind,fcw	0,414	enc,sub	0,165	enc,bcf,sub	0,267
bcf,fcw,sub	0,522	enc,ind,sub	0,406	cff	0,155	sub,sbb	0,264
bcf,fcw,sbb	0,513	enc,ind,cff	0,387	sub	0,151	sbb	0,262
enc,ind,sub	0,491	ind	0,372	cff,sub	0,144	cff,sub,sbb	0,261
bcf,sub,sbb	0,365	enc,bcf,cff	0,37	cff,sub,sbb	0,14	cff,sub	0,259
enc,bcf,sbb	0,354	bcf,cff	0,367	cff,sbb	0,135	sub	0,257
enc,fcw,sbb	0,348	enc,bcf	0,365	sbb	0,13	cff	0,255
bcf,cff,sbb	0,341	ind,sub	0,365	sub,sbb	0,13	cff,sbb	0,255

Также в таблице 7 представлены результаты работ [20, 21], где оценка эффективности одиночных обфусцирующих преобразований и преобразования all выполнена с привлечением символьного

исполнения (столбец  $e_{SE}$ ). Отметим, что для одиночных преобразований рейтинг с учетом порядка совпадает с рейтингом без учета порядка:  $\rho_{+,i}(n) = \rho_{-,i}(n)$  для всех  $n \in \mathcal{O}$ . Для пар и троек преобразований результаты вычисления рейтинга показаны в таблицах 8 и 9 соответственно. Преобразования в таблицах 7–9 выстроены в порядке убывания значений  $\rho_{+,i}, i = 1, 2$ . Заметим, что в работе рассматриваются последовательности длины не более три, и для троек преобразований также выполняется равенство  $\rho_{+,i}(n) = \rho_{-,i}(n)$ .

Таблица 7. Рейтинг эффективности *одиночных* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		[20, 21]	
$n$	$\rho_{\pm,1}$	$n$	$\rho_{\pm,2}$	$n$	$e_{SE}$
all	0,797	all	0,680	all	0,72
ind	0,614	ind	0,454	ind	0,30
bcf	0,457	bcf	0,445	bcf	0,11
fcw	0,457	sbb	0,397	sbb	0,08
sbb	0,361	fcw	0,382	enc	0,08
enc	0,353	enc	0,362	fcw	0,03
sub	0,342	cff	0,343	cff	0,02
cff	0,334	sub	0,33	sub	0,01

Таблица 8. Рейтинг эффективности *пар* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$n$	$\rho_{+,1}$	$n$	$\rho_{+,2}$	$n$	$\rho_{-,1}$	$n$	$\rho_{-,2}$
ind,fcw	0,709	bcf,ind	0,522	ind,fcw	0,709	bcf,sbb	0,528
bcf,ind	0,649	ind,sbb	0,506	ind,bcf	0,649	ind,bcf	0,522
ind,sub	0,598	bcf,sbb	0,505	ind,sbb	0,641	ind,sbb	0,5
ind,sbb	0,597	ind,fcw	0,479	ind,sub	0,589	bcf,fcw	0,487
enc,ind	0,586	bcf,fcw	0,465	bcf,fcw	0,585	ind,fcw	0,479
ind,cff	0,561	fcw,sbb	0,435	ind,enc	0,582	ind,enc	0,449
bcf,fcw	0,542	enc,ind	0,43	ind,cff	0,571	ind,cff	0,417
fcw,sbb	0,49	ind,sub	0,425	bcf,sub	0,424	bcf,enc	0,416
enc,bcf	0,401	enc,bcf	0,416	enc,fcw	0,419	ind,sub	0,416
fcw,sub	0,388	ind,cff	0,415	fcw,sbb	0,415	bcf,cff	0,415
fcw,cff	0,372	bcf,cff	0,385	bcf,sbb	0,411	fcw,sbb	0,384
bcf,sbb	0,345	cff,sbb	0,369	bcf,enc	0,401	enc,sbb	0,372
bcf,sub	0,323	bcf,sub	0,352	bcf,cff	0,393	enc,fcw	0,358
enc,fcw	0,304	sub,sbb	0,344	cff,fcw	0,372	bcf,sub	0,357
bcf,cff	0,303	fcw,cff	0,339	fcw,sub	0,37	cff,sbb	0,351
cff,sbb	0,302	fcw,sub	0,332	enc,sbb	0,308	sbb,sub	0,344
sub,sbb	0,279	enc,fcw	0,31	sbb,sub	0,279	cff,fcw	0,339
cff,sub	0,267	enc,sbb	0,31	cff,sbb	0,275	fcw,sub	0,323
enc,sbb	0,178	enc,sub	0,302	cff,enc	0,274	cff,enc	0,322
enc,cff	0,175	enc,cff	0,29	enc,sub	0,267	enc,sub	0,311
enc,sub	0,167	cff,sub	0,285	cff,sub	0,267	cff,sub	0,285

Таблица 9. Рейтинг эффективности *троек* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$n$	$\rho_{\pm,1}$	$n$	$\rho_{\pm,2}$	$n$	$\rho_{\pm,1}$	$n$	$\rho_{\pm,2}$
ind,fcw,sbb	0.850	bcf,ind,fcw	0.601	enc,ind,sub	0.491	enc,ind,sub	0.406
bcf,ind,fcw	0.802	bcf,fcw,sbb	0.583	bcf,sub,sbb	0.365	enc,ind,cff	0.387
enc,ind,fcw	0.735	bcf,ind,sbb	0.574	enc,bcf,sbb	0.354	enc,bcf,cff	0.370
bcf,ind,sub	0.732	enc,bcf,ind	0.544	enc,fcw,sbb	0.348	ind,cff,sub	0.364
enc,ind,sbb	0.626	ind,fcw,sbb	0.531	bcf,cff,sbb	0.341	bcf,fcw,sub	0.358
ind,fcw,cff	0.617	bcf,cff,sbb	0.528	bcf,cff,sub	0.324	enc,fcw,sbb	0.335
bcf,ind,cff	0.611	enc,bcf,sbb	0.508	enc,bcf,sub	0.313	enc,fcw,sub	0.309
ind,sub,sbb	0.611	bcf,ind,cff	0.486	enc,bcf,cff	0.308	enc,cff,sbb	0.306
ind,fcw,sub	0.600	ind,sub,sbb	0.476	enc,fcw,sub	0.291	enc,sub,sbb	0.304
ind,cff,sbb	0.590	ind,cff,sbb	0.473	fcw,cff,sub	0.277	enc,fcw,cff	0.303
enc,bcf,ind	0.564	bcf,sub,sbb	0.472	enc,fcw,cff	0.272	fcw,sub,sbb	0.285
enc,bcf,fcw	0.563	enc,ind,sbb	0.471	fcw,cff,sbb	0.268	fcw,cff,sbb	0.281
bcf,ind,sbb	0.559	bcf,fcw,cff	0.465	fcw,sub,sbb	0.259	fcw,cff,sub	0.279
ind,cff,sub	0.541	enc,ind,fcw	0.462	enc,cff,sbb	0.175	enc,cff,sub	0.277
enc,ind,cff	0.538	bcf,ind,sub	0.453	enc,cff,sub	0.174	bcf,cff,sub	0.273
bcf,fcw,cff	0.534	enc,bcf,fcw	0.445	enc,sub,sbb	0.169	enc,bcf,sub	0.267
bcf,fcw,sub	0.522	ind,fcw,sub	0.434	cff,sub,sbb	0.140	cff,sub,sbb	0.261
bcf,fcw,sbb	0.513	ind,fcw,cff	0.430				

Для типов последовательностей определим отношения вхождения  $\tau$  в  $\theta(\text{Obf})$  с учетом ( $\in_+$ ) и без учета порядка ( $\in_-$ ) также, как такие отношения определены выше для последовательностей. Тогда для типа  $\tau$  его рейтинг при заданных  $\Delta \in \{+, -\}$  и  $i = 1, 2$  определим формулой:

$$\rho_{\Delta,i}^T(\tau) = \frac{\sum_{\text{Obf} \in Q_{\Delta}^T(\tau)} e_i(\text{Obf})}{|Q_{\Delta}^T(\tau)|},$$

где  $Q_{\Delta}^T(\tau)$  – множество обфусцирующих последовательностей из  $\mathcal{Q} \setminus \{\text{all}\}$ , содержащих преобразования типа  $\tau$ :

$$Q_{\Delta}^T(\tau) = \{\text{Obf} \in \mathcal{Q} \setminus \{\text{all}\} : \tau \in_{\Delta} \theta(\text{Obf})\}. \quad (7)$$

Результаты вычисления рейтинга для одиночных типовых, пар типов и троек типов приведены соответственно в таблицах 10, 11 и 12.

Таблица 10. Рейтинг эффективности типов *одиночных* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		[20, 21]	
$\tau$	$\rho_{\pm,1}^T$	$\tau$	$\rho_{\pm,2}^T$	$\tau$	$e_{SE}$
C	0,446	C	0,401	C	0.143
A	0,408	A	0,390	A	0.050
D	0,360	D	0,352	D	0.045

Таблица 11. Рейтинг эффективности типов *пар* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$\tau$	$\rho_{+,1}^T$	$\tau$	$\rho_{+,2}^T$	$\tau$	$\rho_{-,1}^T$	$\tau$	$\rho_{-,2}^T$
CA	0.522	CA	0.455	CC	0.528	CC	0.447
CC	0.521	CC	0.447	CA	0.477	CA	0.431
AA	0.490	AA	0.435	CD	0.416	AA	0.384
DC	0.419	DC	0.394	AA	0.415	CD	0.375
AD	0.388	CD	0.344	AD	0.359	AD	0.357
CD	0.381	AC	0.339	DD	0.267	DD	0.311
AC	0.373	AD	0.332				
DA	0.279	DA	0.329				
DD	0.167	DD	0.303				

Таблица 12. Рейтинг эффективности типов *троек* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$\tau$	$\rho_{+,1}^T$	$\tau$	$\rho_{+,2}^T$	$\tau$	$\rho_{-,1}^T$	$\tau$	$\rho_{-,2}^T$
CAA	0.681	CAA	0.557	CCC	0.611	CCA	0.512
CCC	0.611	CCA	0.544	CCA	0.574	CCC	0.486
CAC	0.576	CCC	0.486	CAA	0.544	CAA	0.465
CCA	0.573	CAC	0.447	CCD	0.501	CCD	0.398
CAD	0.561	DCA	0.438	CAD	0.437	CAD	0.398
CCD	0.532	DCC	0.434	CDD	0.326	CDD	0.317
DCA	0.491	CDA	0.403	AAD	0.303	AAD	0.310
DCC	0.470	CAD	0.396	ADD	0.230	ADD	0.307
CDA	0.372	CCD	0.363				
DAA	0.348	DAA	0.335				
DCD	0.326	DCD	0.317				
DAD	0.291	DAD	0.309				
ACD	0.277	DDA	0.304				
DAC	0.272	DAC	0.303				
ACA	0.268	ADA	0.285				
ADA	0.259	ACA	0.281				
DDA	0.169	ACD	0.279				

Отметим, что в таблицах 7-12, кроме упорядочения результатов по убыванию рейтинга для наборов  $\mathcal{F}_1$  и  $\mathcal{F}_2$ , строки этих таблиц сгруппированы специальным образом. Группирование выполнено отдельно для случая, когда порядок применения преобразований учитывается, и для случая, когда такой порядок не учитывается. В каждом из этих случаев столбец с названиями наборов преобразований для  $\mathcal{F}_2$  может быть получен путем перестановки строк из столбца с названиями наборов преобразований для  $\mathcal{F}_1$ . Строки таблиц сгруппированы так, что каждая из групп имеет наименьшее число строк и при этом строки для

$\mathcal{F}_2$  могут быть получены из строк для  $\mathcal{F}_1$  путем перестановок строк внутри групп. Такое упорядочение будем использовать для оценки зависимости эффективности преобразований от набора программ. Степень зависимости определим как среднее число строк в группе (результат деления общего числа строк на число групп). Результаты вычисления степени зависимости, полученные по таблицам 7-12, приведены в таблице 13, откуда видно, что с ростом числа преобразований эта зависимость проявляется больше, как в случае учета порядка преобразований, так и без учета.

Таблица 13. Оценка степени зависимости эффективности от набора программ:

$|n|$  и  $|\theta(n)|$  – соответственно число базовых преобразований и число типов в наборе  $n \in \mathcal{O}^i$ ,  $|n| = |\theta(n)| = i$

$\Delta$	$ n $			$ \theta(n) $		
	1	2	3	1	2	3
+	1, 4	10, 5	17, 5	1	$\approx 1, 28$	$\approx 2, 83$
-					1, 2	$\approx 1, 14$

### 6.3. Оценка эффективности обфусцирующих преобразований.

Анализ таблиц 7-13, помимо выводов о соотношении эффективности конкретных преобразований и типов преобразований, позволяет сделать следующие выводы.

Результаты оценки эффективности с помощью предложенных функций похожести (на основе статических признаков) коррелируют с результатами работ [20, 21], где оценка эффективности выполнена с использованием символического исполнения (на основе динамических признаков). В частности, для различных подходов к оценке эффективности, согласно, таблице 7, преобразования `ind` и `bcf`, демонстрирует наибольшую эффективность, а преобразования `sub` и `cff` – наименьшую. Отметим, что особое внимание к преобразованию `bcf` (непрозрачные или неявные предикаты), например, уделяемое в диссертациях [53] и [6], подтверждает результаты настоящей работы. Более того, для множества  $\mathcal{F}_2$  упорядоченный набор преобразований практически совпадает с упорядоченным набором преобразований, полученным в [20, 21]; отличие заключается только в порядке следования соседних преобразований `eps` и `fcw`. Отметим, что множество  $\mathcal{F}_2$  построено на основе набора программ [48], который использовался и в [20, 21]. Поэтому близость результатов оценки эффективности при различных подходах их получения может свидетельствовать в пользу достоверности этих результатов.

Эффективность преобразования зависит от выбранной функции похожести в модели (5). Наибольшей эффективностью для построенных в работе функций похожести обладают преобразования типа С, которые направлены на изменение графа потока управления (таблица 10), а наименьшей эффективностью – преобразования типа D. Высокая эффективность преобразований типа С подтверждается независимыми исследованиями, например, [52]. Заметим, что в [16] получен обратный результат: обфусцирующие преобразования над данными оказываются более эффективными, чем преобразования графа потока управления. Причина такого различия, как представляется, заключается в использовании разных функций похожести. В [16] эффективность оценивается с помощью сложности программы по Колмогорову, которая аппроксимируется сжимаемостью программы алгоритмами сжатия. Преобразования данных, в частности, шифрование, в большей степени, чем другие преобразования, делают программы более похожими на наборы случайных данных, что приводит к меньшей сжимаемости программ, и объясняет их высокую сложность по Колмогорову. В настоящей работе для построения функции похожести помимо степени сжатия используются специализированные характеристики, учитывающие структуру и характер данных исполняемого файла. Представляется, что демонстрируемая в настоящем исследовании высокая эффективность преобразований типа С объясняется тем, что построенные функции похожести используют, в том числе, информацию о графе потока управления. Другими словами, преобразования типа С меняют граф программы  $P$ , делая его менее похожим на граф программы  $A(P_0)$ . Другими словами, эффективность обфусцирующих преобразований зависит от используемой функции похожести, которую можно рассматривать как модель аналитика, против которого выбирается эффективное защитное обфусцирующее преобразование. Тем не менее, влияние обфусцирующих преобразований типа С на эффективность обфусцирующей последовательности независимо подтверждается особым вниманием исследователей к таким преобразованиям (например, [6, 13, 28, 50, 53 – 56]). Такой интерес может быть обоснован тем, что часто аналитик начинает исследование программы с построения графа потока управления [57], поэтому естественным направлением исследований в области защиты программ является затруднение восстановления такого графа. При этом стоит отметить, что не все преобразования этого типа могут обеспечивать высокую эффективность. В частности, из таблиц 7-9 видно, что группы, содержащие наименее эффективные наборы, содержат наборы, в которые из преобразований типа С входит

только преобразование cff. Слабая эффективность этого преобразования показана с применением техники символьного исполнения в [20, 21] (крайние правые два столбца в таблице 7) и независимо показана в [51] также с помощью символьного исполнения.

Эффективность обфусцирующего преобразования зависит от программы, к которой оно применяется, и порядка применения базовых преобразований. В частности, согласно таблице 7, на наборе  $\mathcal{F}_1$  преобразование fcw эффективнее sbb, а sub эффективнее cff. В то же время на наборе  $\mathcal{F}_2$  для указанных пар наблюдается обратная ситуация. В таблице 8 эта зависимость проявляется в большей степени для пар базовых преобразований. Например, можно выделить только три пары преобразований, находящихся в первой пятерке наиболее эффективных преобразований и для  $\mathcal{F}_1$  и для  $\mathcal{F}_2$ : (ind,fcw), (bcf,ind) и (ind,sbb). Также можно выделить только два преобразования, которые независимо от набора программ и порядка применения базовых преобразований, находятся в пятерке наименее эффективных:(cff,sub) и (enc,cff). В остальных случаях наблюдается зависимость от порядка применения преобразований и/или набора программ. Для троек преобразований (напомним, что в этом случае  $\rho_{+,i} = \rho_{-,i}$ ), как следует из таблицы 12, в первой пятерке наиболее эффективных троек независимо от рассматриваемого набора находятся преобразования (ind,fcw,sbb) и (bcf,ind,fcw). Независимость от набора программ демонстрирует также наименее эффективные преобразования (cff,sub,sbb) и (enc,cff,sub). В остальных случаях для троек наблюдается зависимость от набора программ. Степень зависимости эффективности от набора программ отражена в таблице 13. Особенно эта зависимость проявляется с ростом числа преобразований в последовательности. Для типов преобразований зависимость от программ проявляется в меньшей степени. В большей степени наблюдается зависимость от порядка и типа применяемых преобразований (следующий вывод).

Эффективность зависит от типов преобразования и порядка следования этих типов. Согласно таблицам 11 и 12, комбинации преобразований типа С и А позволяют строить эффективные обфусцирующие последовательности, причем эффективность зависит от порядка применения типов: последовательности, в которых сначала применяется преобразование типа С, а затем – типа А, существенно эффективнее обратного порядка. Этот эффект может быть объяснен тем, что преобразование типа А может изменить программу так, что применение преобразования С становится затруднительным. Результаты таблицы 12, в которых учитывается порядок применения преобразований,

можно условно разделить на две части: первые 9 строк, последние 8 строк. В первой части, как для  $\mathcal{F}_1$ , так и для  $\mathcal{F}_2$ , в каждом из типов на первом или втором месте есть преобразование типа С, причем нет типов, начинающихся типом А, и тип А всегда следует после типа С. Во второй части, как для  $\mathcal{F}_1$ , так и для  $\mathcal{F}_2$ , расположены типы, либо начинающиеся типом А, либо начинающиеся D, в которых тип А, при наличии, следует после D, либо начинающиеся D и содержащие два преобразования типа D. Таким образом, множество из первых 9-ти типов преобразований для набора  $\mathcal{F}_1$  совпадает с точностью до перестановки со множеством из первых 9-ти типов преобразований для набора  $\mathcal{F}_2$ . Аналогичный вывод справедлив и для типов из второй части. Последовательности, в которых не используются преобразования типа С, обладают наименьшей эффективностью. Отметим, что в соответствии с результатами работы [8] в среднем преобразования типов CDA и DAC являются *дефективными* (время символьного анализа уменьшилось по сравнению со временем символьного исполнения оригинальной программы), что согласуется с полученными в настоящей работе результатами. Именно, не смотря на то, что CDA относится к первой части, оно занимает последнее 9-е место для набора  $\mathcal{F}_1$  и 7-е место для набора  $\mathcal{F}_2$ . Отметим, что в типах CDA и DAC преобразование типа А идет после преобразования типа D, что выше для результатов, полученных в этой работе, отмечено как признак малоэффективных преобразований. Тип DAC относится ко второй части и занимает в этой части 5-е место среди восьми для обоих наборов программ. Преобразования CAD и DCA содержатся на пятом месте первой части для обоих наборов и в соответствии с [18] также являются эффективными.

В [18] не исследовались пары преобразований, а также тройки преобразований вида  $(X, Y, Z)$ , в которых преобразования  $X$  и  $Z$  одного типа разделяются преобразованием  $Y$  другого типа. Однако результаты показывают, что такие преобразования могут быть более эффективными. В частности, для набора  $\mathcal{F}_1$  наибольшую эффективность показало преобразование  $(\text{ind}, \text{fcw}, \text{sbb})$  (последовательность типа САА).

Увеличение числа преобразований в обфусцирующих последовательностях по-разному влияет на изменение эффективности (уменьшение похожести). В частности, на наборе  $\mathcal{F}_1$  добавление преобразования  $\text{sff}$  к последовательностям  $X$  и  $(X, Y)$  приводит к уменьшению эффективности в 8-ми из 10-ти случаев. Добавление преобразования  $\text{sub}$  к последовательностям  $X$  и  $(X, Y)$  в 8-ми из 15-ти случаев приводит к уменьшению эффективности. Добавление преобразования  $\text{sbb}$  к последовательностям  $X$  и  $(X, Y)$  в 8-ми из 21-ого



случаев приводит к уменьшению эффективности. На множестве  $\mathcal{F}_2$  в 5-ти из 10-ти случаев добавление преобразования cff к последовательностям  $X$  и  $(X,Y)$  приводит к уменьшению эффективности. Добавление преобразования sub к последовательностям  $X$  и  $(X,Y)$  в 10-ти из 15-ти случаев приводит к уменьшению эффективности. Добавление преобразования sbb к последовательностям  $X$  и  $(X,Y)$  только в одном случае из 21-го приводит к уменьшению эффективности.

Имеется группа преобразований: {cff, sub, (cff,sub), (cff,sub,sbb), (cff,sbb), sbb, (cff,sbb), sbb, (sub,sbb)}, которые показывают наименьшую эффективность на обоих наборах  $\mathcal{F}_1$  и  $\mathcal{F}_2$ . То есть можно отметить обфусцирующие преобразования, которые, как представляется, обладают малой эффективностью независимо от обфусцируемой программы. Аналогичный вывод можно сделать и для наиболее эффективных преобразований {all, (bcf,ind,fcw)}, которые демонстрируют наибольшую эффективность на наборах  $\mathcal{F}_1$  и  $\mathcal{F}_2$ . В остальных случаях наблюдается зависимость эффективности от обфусцируемой программы.

#### 6.4. Исследование стойкости обфусцирующих преобразований.

Для оценки стойкости реализована схема, изображенная на рисунке 1. В качестве деобфускатора применен оптимизатор компилятора Clang для IR-кода (параметр оптимизации Os). Отметим, что в работах [20, 21] в качестве деобфускатора при оценке стойкости использовался такой же оптимизатор. В таблице 14 для каждого из  $\text{Obf} \in \mathcal{Q}$  в столбцах с именами  $r_1$  и  $r_2$  приведено среднее значение стойкости, вычисленное по формуле (4) на основе функций похожести  $s_c$  для всех классификаторов  $c$  из таблицы 3. Первый и третий столбец таблицы содержит преобразования, упорядоченные по убыванию значения стойкости  $r_1$  (от более эффективных к менее эффективным, или, что то же самое, от менее похожих на аппроксимацию к более похожим), полученного на множестве  $\mathcal{F}_1$ , а второй и четвертый – преобразования, упорядоченные по убыванию значения  $r_2$ , полученного на множестве  $\mathcal{F}_2$ .

По аналогии с рейтингом эффективности для заданных  $\mathcal{F}_i, i = 1, 2$ , и  $\Delta \in \{+, -\}$  определим рейтинг стойкости одиночных преобразований, пар преобразований, троек преобразований и соответствующих комбинаций типов преобразований:

$$\varrho_{\Delta,i}(n) = \frac{\sum_{\text{Obf} \in Q_{\Delta}(n)} r_i(\text{Obf}, D)}{|Q_{\Delta}(n)|},$$

$$\varrho_{\Delta,i}^T(t) = \frac{\sum_{\text{Obf} \in Q_{\Delta}^T(t)} r_i(\text{Obf}, D)}{|Q_{\Delta}^T(t)|},$$

где символом  $r_i(\text{Obf}, D)$  здесь обозначается стойкость  $r_i$  из таблицы 14 преобразования Obf по отношению к деобфускатору D, а множества  $Q_{\Delta}(n)$  и  $Q_{\Delta}^T(n)$  определены соответственно в (6) и (7). Результаты вычисления рейтинга стойкости, упорядоченные по убыванию стойкости, представлены в таблицах 15,16,17,18,19,20. В таблице 21 представлены результаты оценки зависимости значений стойкости для выбранных функций похожести от используемого тестового набора программ. Как и для эффективности такая зависимость определяется как среднее число строк в группе.

Таблица 14. Упорядоченные по убыванию значений стойкости  $r_1$  и  $r_2$  обфусцирующие преобразования длины один, два и три

$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
Obf	$r_1$	Obf	$r_2$	Obf	$r_1$	Obf	$r_2$
all	0,905	all	0,585	fcw,sub,sbb	0,276	enc,cff,sub	0,288
ind,fcw,sbb	0,883	bcf,ind,fcw	0,567	fcw,cff,sbb	0,272	enc,cff,sbb	0,288
bcf,ind,fcw	0,869	bcf,ind,sbb	0,545	fcw,cff,sub	0,269	fcw,sub,sbb	0,284
bcf,ind,sbb	0,766	ind,fcw,sbb	0,505	bcf,fcw,sub	0,267	fcw,cff	0,282
bcf,ind	0,703	bcf,ind	0,5	fcw,cff	0,262	fcw,sbb	0,281
bcf,ind,cff	0,692	enc,bcf,ind	0,497	bcf,sub,sbb	0,261	fcw	0,276
ind,sbb	0,689	bcf,ind,sub	0,495	enc,cff,sbb	0,253	fcw,sub	0,276
enc,bcf,ind	0,675	enc,ind,sbb	0,493	enc,sub,sbb	0,252	fcw,cff,sbb	0,276
bcf,ind,sub	0,675	bcf,ind,cff	0,491	bcf,cff,sbb	0,244	sbb	0,274
ind,cff,sbb	0,657	ind,cff,sbb	0,479	enc,bcf,sbb	0,244	cff,sub	0,273
ind,sub,sbb	0,65	ind,sbb	0,478	fcw	0,243	fcw,cff,sub	0,271
enc,ind,fcw	0,65	ind,sub,sbb	0,475	fcw,sub	0,242	cff,sub,sbb	0,271
ind,fcw,sub	0,649	enc,ind,fcw	0,454	enc	0,24	cff	0,27
ind,fcw,cff	0,637	ind,fcw	0,433	enc,sbb	0,238	sub,sbb	0,269
ind,fcw	0,63	ind,fcw,sub	0,432	enc,cff	0,235	cff,sbb	0,266
enc,ind,sbb	0,58	ind,fcw,cff	0,431	bcf,sbb	0,228	enc,bcf,cff	0,265
ind	0,519	enc,ind	0,411	enc,bcf,cff	0,213	enc,bcf,fcw	0,259
ind,sub	0,514	enc,ind,cff	0,409	bcf	0,206	sub	0,259
ind,cff,sub	0,512	ind,cff,sub	0,394	cff,sub	0,205	bcf,fcw	0,257
ind,cff	0,505	enc,ind,sub	0,391	sub	0,203	enc,bcf	0,256
enc,ind,sub	0,49	ind,cff	0,391	cff	0,203	enc,bcf,sub	0,256
enc,ind	0,449	ind,sub	0,385	bcf,cff,sub	0,196	bcf,fcw,sub	0,253
enc,ind,cff	0,438	ind	0,382	enc,bcf	0,194	bcf,sub	0,253
enc,fcw,sbb	0,43	enc,fcw,sbb	0,316	bcf,cff	0,189	bcf	0,253
enc,bcf,fcw	0,375	enc,sbb	0,308	bcf,sub	0,188	bcf,cff	0,251
enc,fcw,cff	0,34	enc,sub,sbb	0,306	cff,sub,sbb	0,186	bcf,fcw,cff	0,244
bcf,fcw,sbb	0,315	enc,fcw,cff	0,297	enc,sub	0,184	bcf,fcw,sbb	0,238
bcf,fcw,cff	0,303	enc,fcw	0,295	enc,cff,sub	0,175	bcf,cff,sub	0,237
enc,fcw	0,29	enc,cff	0,294	sbb	0,174	bcf,cff,sbb	0,235
bcf,fcw	0,287	enc	0,293	cff,sbb	0,17	enc,bcf,sbb	0,229
enc,fcw,sub	0,277	enc,fcw,sub	0,292	sub,sbb	0,17	bcf,sub,sbb	0,219
fcw,sbb	0,277	enc,sub	0,291	enc,bcf,sub	0,168	bcf,sbb	0,21

Таблица 15. Рейтинг стойкости *одиночных* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		[20, 21]	
$n$	$\varrho_{\pm,1}$	$n$	$\varrho_{\pm,2}$	$n$	$r_{SE}$
all	0,905	all	0,585	all	0.709
ind	0,629	ind	0,456	ind	0.300
fcw	0,411	sbb	0,329	bcf	0.103
bcf	0,376	fcw	0,328	sbb	0.075
sbb	0,373	enc	0,327	enc	0.070
enc	0,336	bcf	0,319	fcw	0.025
cff	0,325	cff	0,314	cff	0.022
sub	0,319	sub	0,312	sub	0.019

Таблица 16. Рейтинг стойкости *пар* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$n$	$\varrho_{+,1}$	$n$	$\varrho_{+,2}$	$n$	$\varrho_{-,1}$	$n$	$\varrho_{-,2}$
bcf,ind	0,732	bcf,ind	0,516	ind,bcf	0,732	ind,bcf	0,516
ind,fcw	0,720	ind,sbb	0,505	ind,fcw	0,720	ind,sbb	0,496
ind,sbb	0,678	ind,fcw	0,470	ind,sbb	0,702	ind,fcw	0,470
ind,sub	0,582	ind,sub	0,436	ind,sub	0,582	ind,enc	0,443
ind,cff	0,561	ind,cff	0,433	ind,cff	0,574	ind,cff	0,433
enc,ind	0,521	enc,ind	0,432	ind,enc	0,547	ind,sub	0,429
fcw,sbb	0,473	fcw,sbb	0,335	fcw,sbb	0,407	enc,sbb	0,323
fcw,cff	0,347	cff,sbb	0,309	bcf,fcw	0,405	enc,fcw	0,319
fcw,sub	0,342	enc,sbb	0,308	enc,fcw	0,394	fcw,sbb	0,317
enc,fcw	0,334	fcw,sub	0,307	cff,fcw	0,347	cff,enc	0,307
cff,sbb	0,319	sub,sbb	0,304	bcf,sbb	0,343	enc,sub	0,304
enc,bcf	0,312	fcw,cff	0,300	enc,sbb	0,333	sbb,sub	0,304
bcf,fcw	0,309	enc,fcw	0,300	fcw,sub	0,330	bcf,fcw	0,303
sub,sbb	0,299	enc,sub	0,299	bcf,enc	0,312	cff,sbb	0,302
cff,sub	0,257	enc,bcf	0,294	bcf,cff	0,306	fcw,sub	0,301
enc,sbb	0,238	enc,cff	0,290	sbb,sub	0,299	cff,fcw	0,300
bcf,sbb	0,236	cff,sub	0,289	cff,sbb	0,297	bcf,enc	0,294
enc,cff	0,221	bcf,fcw	0,250	bcf,sub	0,292	cff,sub	0,289
enc,sub	0,218	bcf,cff	0,247	cff,enc	0,276	bcf,cff	0,287
bcf,cff	0,210	bcf,sub	0,243	enc,sub	0,258	bcf,sub	0,286
bcf,sub	0,206	bcf,sbb	0,219	cff,sub	0,257	bcf,sbb	0,279

### 6.5. Оценка стойкости обфусцирующих преобразований.

Анализ результатов в таблицах, помимо выводов о соотношении рейтинга стойкости отдельных наборов преобразований и их типов, позволяет сделать следующие выводы.

Таблица 17. Рейтинг стойкости *троек* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$n$	$\varrho_{\pm,1}$	$n$	$\varrho_{\pm,2}$	$n$	$\varrho_{\pm,1}$	$n$	$\varrho_{\pm,2}$
bcf,ind,fcw	0,883	bcf,ind,fcw	0,567	bcf,fcw,sbb	0,315	enc,fcw,sub	0,292
ind,fcw,sbb	0,869	bcf,ind,sbb	0,545	bcf,fcw,cff	0,303	enc,cff,sub	0,288
bcf,ind,sbb	0,766	ind,fcw,sbb	0,505	enc,fcw,sub	0,277	enc,cff,sbb	0,288
bcf,ind,cff	0,692	enc,bcf,ind	0,497	fcw,sub,sbb	0,276	fcw,sub,sbb	0,284
bcf,ind,sub	0,675	bcf,ind,sub	0,495	fcw,cff,sbb	0,272	fcw,cff,sbb	0,276
enc,bcf,ind	0,675	enc,ind,sbb	0,493	fcw,cff,sub	0,269	fcw,cff,sub	0,271
ind,cff,sbb	0,657	bcf,ind,cff	0,491	bcf,fcw,sub	0,267	cff,sub,sbb	0,271
ind,sub,sbb	0,650	ind,cff,sbb	0,479	bcf,sub,sbb	0,261	enc,bcf,cff	0,265
enc,ind,fcw	0,650	ind,sub,sbb	0,475	enc,cff,sbb	0,253	enc,bcf,fcw	0,259
ind,fcw,sub	0,649	enc,ind,fcw	0,454	enc,sub,sbb	0,252	enc,bcf,sub	0,256
ind,fcw,cff	0,637	ind,fcw,sub	0,432	enc,bcf,sbb	0,244	bcf,fcw,sub	0,253
enc,ind,sbb	0,580	ind,fcw,cff	0,431	bcf,cff,sbb	0,244	bcf,fcw,cff	0,244
ind,cff,sub	0,512	enc,ind,cff	0,409	enc,bcf,cff	0,213	bcf,fcw,sbb	0,238
enc,ind,sub	0,490	ind,cff,sub	0,394	bcf,cff,sub	0,196	bcf,cff,sub	0,237
enc,ind,cff	0,438	enc,ind,sub	0,391	cff,sub,sbb	0,186	bcf,cff,sbb	0,235
enc,fcw,sbb	0,430	enc,fcw,sbb	0,316	enc,cff,sub	0,175	enc,bcf,sbb	0,229
enc,bcf,fcw	0,375	enc,sub,sbb	0,306	enc,bcf,sub	0,168	bcf,sub,sbb	0,219
enc,fcw,cff	0,340	enc,fcw,cff	0,297				

Таблица 18. Рейтинг стойкости типов *одиночных* обфусцирующих преобразований

$\mathcal{F}_1$		$\mathcal{F}_2$		[20, 21]	
$t$	$\varrho_{\pm,1}^T$	$t$	$\varrho_{\pm,2}^T$	$t$	$r_{SE}$
C	0,414	C	0,348	C	0,208
A	0,390	A	0,331	A	0,050
D	0,338	D	0,322	D	0,045

Таблица 19. Рейтинг стойкости типов *пар* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$t$	$\varrho_{+,1}^T$	$t$	$\varrho_{+,2}^T$	$t$	$\varrho_{-,1}^T$	$t$	$\varrho_{-,2}^T$
CC	0,525	CC	0,411	CC	0,518	CC	0,402
CA	0,475	CA	0,361	CA	0,442	CA	0,348
AA	0,473	DC	0,342	AA	0,407	CD	0,335
DC	0,367	AA	0,335	CD	0,372	AA	0,317
AC	0,347	CD	0,324	AD	0,345	AD	0,314
CD	0,345	AD	0,307	DD	0,258	DD	0,304
AD	0,342	DA	0,303				
DA	0,306	AC	0,300				
DD	0,218	DD	0,299				

Рейтинг стойкости одиночных преобразований, представленный в таблицах 15 и 18 и полученный на основе статических характеристик программ, частично коррелирует с результатами оценки стойкости

этих преобразований, полученными в [20, 21] на основе динамических характеристик – характеристик символического исполнения. В частности, результаты, полученные в настоящей работе, также как и результаты работ [20, 21] позволяют в качестве наиболее стойкого выделить преобразование ind, а в качестве наименее стойких – преобразования sff и sub (таблица 15). Относительно других одиночных преобразований (bcf, sbb, enc и fcw) корреляция явно не выражена: только в случае множества  $\mathcal{F}_2$  номер строки в таблице 15, где расположено преобразование enc, совпадает с номером строки этого преобразования, когда оценка стойкости выполнена методами символического исполнения. В то же время, среди обфусцирующих последовательностей, содержащих только одно преобразование из базового набора, наибольшей стойкостью в среднем обладают преобразования типа C, а наименьшей – преобразования типа D (таблица 18). Эти результаты качественно совпадают с усредненными результатами оценки стойкости типов преобразований, полученными в [20, 21]: набор типов одиночных преобразований, упорядоченный по убыванию усредненного рейтинга стойкости, полученный в настоящей работе, совпадает с аналогичным образом упорядоченным набором типов, полученным на основе результатов [20, 21] (таблица 18). Также стоит отметить, что референсное преобразование all во всех случаях обладает наибольшим рейтингом.

Таблица 20. Рейтинг стойкости типов *троек* обфусцирующих преобразований

С учетом порядка				Без учета порядка			
$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_1$		$\mathcal{F}_2$	
$t$	$\varrho_{+,1}^T$	$t$	$\varrho_{+,2}^T$	$t$	$\varrho_{-,1}^T$	$t$	$\varrho_{-,2}^T$
CCC	0,692	CCC	0,491	CCC	0,692	CCC	0,491
CCA	0,637	CCA	0,456	CCA	0,582	CCA	0,417
CAA	0,592	DCC	0,390	CAA	0,485	CCD	0,383
CAC	0,470	CCD	0,375	CCD	0,452	CAA	0,340
CCD	0,461	CAA	0,371	CAD	0,394	CAD	0,328
CAD	0,458	DCA	0,345	AAD	0,353	CDD	0,312
DCC	0,442	CAD	0,343	CDD	0,278	AAD	0,300
DAA	0,430	CAC	0,337	ADD	0,265	ADD	0,299
DCA	0,420	CDA	0,322				
CDA	0,366	DAA	0,316				
DAC	0,340	DCD	0,312				
DCD	0,278	DDA	0,306				
DAD	0,277	DAC	0,297				
ADA	0,276	DAD	0,292				
ACA	0,272	ADA	0,284				
ACD	0,269	ACA	0,276				
DDA	0,252	ACD	0,271				

Таблица 21. Оценка степени зависимости стойкости от набора программ:  $|n|$  и  $|\theta(n)|$  – соответственно число базовых преобразований и число типов в наборе

$$n \in \mathcal{O}^i, |n| = |\theta(n)| = i$$

$\Delta$	$ n $			$ \theta(n) $		
	1	2	3	1	2	3
+		3		1,8	4,25	
-	1,75	5,25	5	1,2	$\approx 2,67$	

Стойкость обфусцирующего преобразования зависит от программы, к которой оно применяется, и порядка применения базовых преобразований. В частности, согласно таблице 15, на наборе  $\mathcal{F}_1$  преобразования fsw и bcf более стойкие, чем sbb и eps соответственно. В то же время на наборе  $\mathcal{F}_2$  для указанных пар наблюдается обратная ситуация. В таблице 16 эта зависимость проявляется в большей степени для пар базовых преобразований. Например, в левой половине таблицы 16 (оценка стойкости с учетом порядка преобразований) можно выделить только семь пар преобразований, рейтинг которых практически не зависит от набора данных; в правой половине таблицы 16 (оценка стойкости без учета порядка преобразований) можно выделить только шесть таких пар преобразований. В остальных случаях наблюдается зависимость от порядка применения преобразований и/или набора программ. Для троек преобразований (напомним, что в этом случае  $\varrho_{+,i} = \varrho_{-,i}$ ), как следует из таблицы 20, в первой пятёрке наиболее стойких троек независимо от рассматриваемого набора находятся четыре последовательности: (bcf,ind,fcw), (ind,fcw,sbb), (bcf,ind,fcw), (bcf,ind,sub). В остальных случаях для троек наблюдается зависимость от набора программ. Степень зависимости стойкости от набора программ отражена в таблице 21. Стоит отметить, что зависимость стойкости конкретных последовательностей преобразований от набора программ проявляется в меньшей степени, зависимость эффективности этих преобразований (ср. таблицы 13 и 21). Представляется, что применение одного деобфускатора D к программам из разных наборов делает эти наборы «менее различимыми» рассматриваемыми классификаторами из таблицы 3.

Стойкость зависит от типов преобразования и порядка следования этих типов. Согласно таблицам 19 и 20, комбинации преобразований типа C и A позволяют строить стойкие обфусцирующие последовательности, причем стойкость зависит от порядка применения типов: последовательности, в которых сначала применяется преобразование типа C, а затем – типа A, более стойкие, чем последовательности с обратным порядком следования типов. Этот

эффект может быть объяснен тем, что преобразование типа А может изменить программу так, что применение преобразования С становится затруднительным. Результаты таблицы 20, в которых учитывается порядок применения преобразований, можно условно выделить две части: первые 7 строк, последние 7 строк. В первой части, как для  $\mathcal{F}_1$ , так и для  $\mathcal{F}_2$ , в каждом из типов на первом или втором месте есть преобразование типа С, причем нет типов, начинающихся типом А, и тип А всегда следует после типа С. Во второй части, как для  $\mathcal{F}_1$ , так и для  $\mathcal{F}_2$ , расположены типы, либо начинающиеся типом А, либо начинающиеся типом D, в которых тип А, при наличии, следует после D, либо начинающиеся с D и содержащие два преобразования типа D.

Увеличение числа преобразований в обфусцирующих последовательностях по-разному влияет на изменение стойкости. В частности, на наборах  $\mathcal{F}_1$  и  $\mathcal{F}_2$  добавление преобразования `bcf` уменьшает стойкость: при добавлении к преобразованию `enc`). На наборе  $\mathcal{F}_1$  добавление преобразования `sff` к последовательностям  $X$  и  $(X,Y)$  приводит к уменьшению стойкости в 5-ти из 10-ти случаев; добавление преобразования `sub` к последовательностям  $X$  и  $(X,Y)$  в 9-ми из 15-ти случаев приводит к уменьшению стойкости; добавление преобразования `sbb` к последовательностям  $X$  и  $(X,Y)$  в 4-ех из 21-ого случая приводит к уменьшению стойкости. На множестве  $\mathcal{F}_2$  в 5-ти из 10-ти случаев добавление преобразования `sff` к последовательностям  $X$  и  $(X,Y)$  приводит к уменьшению стойкости. Добавление преобразования `sub` к последовательностям  $X$  и  $(X,Y)$  в 11-ти из 15-ти случаев приводит к уменьшению стойкости. Добавление преобразования `sbb` к последовательностям  $X$  и  $(X,Y)$  в 9-ти из 21-го случая приводит к уменьшению стойкости.

**6.6. Общие выводы.** Можно заключить, что в среднем наиболее эффективными и стойкими являются обфусцирующие последовательности, начинающиеся преобразованием типа С, а наименее эффективными – последовательности, начинающиеся преобразованием типа D и при этом не содержащие преобразований типа С, либо последовательности, начинающиеся преобразованием типа А. Стоит отметить, что не все преобразования типа С демонстрируют высокую эффективность и стойкость. Так, преобразование `sff` имеет низкий рейтинг как по эффективности, так и по стойкости: в таблицах 7 и 15 это преобразование занимает предпоследнюю или последнюю строки.

Последовательное добавление преобразований не всегда приводит к увеличению стойкости и эффективности обфусцирующих последовательностей. Результаты экспериментов показывают, что

последовательное добавление преобразований  $ind$  и  $fcw$  (тип  $C$ ) к обфусцирующим последовательностям  $X$  и  $(X,Y)$  всегда приводит к увеличению эффективности и стойкости преобразований. Влияние преобразования  $ind$  на эффективность и стойкость, как представляется, связано с проблемами распознавания косвенных переходов автоматическими средствами статического анализа машинного кода, такими как дизассемблеры [58], что в свою очередь может приводить к ошибкам и/или неточностям восстановления, например, графа потока управления, на основе которого также принимается решение о степени похожести программ (специализированные характеристики в таблице 2). Трудность распознавания адреса косвенных переходов подтверждается теоретическими исследованиями: в [57] доказано, что точное определение адреса для косвенных переходов является NP-трудной задачей. Поэтому в [57] для затруднения восстановления графа потока управления предлагается способ на основе замены прямых переходов косвенными, когда адрес перехода вычисляется динамически. На практике сложность анализа кода, содержащего косвенные переходы, подтверждается также сложностью динамического анализа, а именно сложностью символьной интерпретации кода, содержащего такие переходы [49]. В частности, такое подтверждение получено в [20, 21] (крайние правые два столбца в таблице 7).

**7. Заключение.** Заметим, что в [8] выделено всего 5 работ из 572, в которых предлагаются не только способы защиты, но и проводится анализ случаев, когда противник адаптирует средства анализа для обхода защиты. Малое число работ, учитывающих возможность адаптации противника, подчеркивает актуальность задачи количественной оценки эффективности и стойкости защиты при наличии подходящих средств анализа у противника. Предлагаемый в настоящей работе способ оценки эффективности и стойкости позволяет адаптировать оценку при появлении новых средств анализа. Это может быть выполнено за счет дообучения существующей функции похожести или построения новой функции.

Отметим, что предлагаемый в настоящей работе подход, может быть применен для оценки и эффективности, и стойкости. Представляется, что при подборе подходящей функции похожести этот метод оценки эффективности и стойкости может быть применен не только к бинарным исполняемым файлам, но и к файлам с исходным кодом и файлам с кодом промежуточного представления. Для оценки эффективности преобразований конкретных программ рекомендуется



использовать функции похожести, обладающие высокой точностью, например, функцию  $s_{NN}$  на основе нейронных сетей.

Дальнейшими направлениями исследований являются: 1) построение функций похожести с высокой точностью на наборах, отличных от наборов, использованных при обучении; 2) построение методами машинного обучения функций похожести, для которых возможно дообучение не только с помощью новых наборов данных, но и с помощью новых признаков; 3) построение функций похожести без использования методов машинного обучения, например, построение нечетких хеш-функций для исполняемых файлов.

### Литература

1. Akhunzada A., Sookhak M., Anuar A.B., Gani A., Ahmed E., Shiraz M., Furnell S., Hayat A., Khan M.K. Man-At-The-End attacks: Analysis, taxonomy, human aspects, motivation and future directions // *Journal of Network and Computer Applications*. 2015. vol. 48. pp. 44–57.
2. Undrits R., Resende J. et. al. CyberSec4Europe D3.23: Cybersecurity Outlook 2. Research Report D3.23. 2022. pp. 1–82.
3. Biernacki L., Gallagher M., Xu Z., Aga M.T., Harris A., Wei S., Tiwari M., Kasikci B., Malik S., Austin T. Software-driven security attacks: From vulnerability sources to durable hardware defenses // *ACM Journal on Emerging Technologies in Computing Systems (JETC)*. 2021. vol. 17. no. 3. pp. 1–38.
4. Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.В. Современное состояние исследований в области обфускации программ: определения стойкости обфускации // *Труды Института системного программирования РАН*. 2014. Т. 26. № 3. С. 167–198.
5. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vadhan S., Yang K. On the (im) possibility of obfuscating programs // *Journal of the ACM (JACM)*. 2012. vol. 59. no. 2. pp. 1–42.
6. Zobernig L. Mathematical Aspects of Program Obfuscation. Doctoral dissertation. ResearchSpace@ Auckland, 2020. URL: [www.math.auckland.ac.nz/~sgal018/Lukas-Zobernig-Thesis.pdf](http://www.math.auckland.ac.nz/~sgal018/Lukas-Zobernig-Thesis.pdf) (дата обращения: 19.06.2023).
7. Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B. Candidate indistinguishability obfuscation and functional encryption for all circuits // *SIAM Journal on Computing*. 2016. vol. 45. no. 3. pp. 882–929.
8. Kochberger P., Schrittwieser S., Coppens B., De Sutter B. Evaluation Methodologies in Software Protection Research // *arXiv preprint arXiv:2307.07300*. 2023. pp. 1–67.
9. Zhou Y., Main A., Gu Y.X., Johnson H. Information hiding in software with mixed boolean-arithmetic transforms // *Information Security Applications: 8th International Workshop on Information Security Applications*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007. vol. 4867. pp. 61–75.
10. Reichenwallner B., Meerwald-Stadler P. Efficient Deobfuscation of Linear Mixed Boolean- Arithmetic Expressions // *Proceedings of the 2022 ACM Workshop on Research on offensive and defensive techniques in the context of Man At The End (MATE) attacks*. 2022. pp. 19–28.
11. Xu D., Liu D., Feng W., Ming J., Zheng Q., Li J., Yu Q. Boosting SMT solver performance on mixed-bitwise-arithmetic expressions // *Proceedings of the 42nd ACM SIGPLAN*

- International Conference on Programming Language Design and Implementation. 2021. pp. 651–664.
12. Liu B., Shen J., Ming J., Zheng Q., Li J., Xu D. MBA-Blast: Unveiling and Simplifying Mixed Boolean-Arithmetic Obfuscation // Proceedings of the 30th USENIX Security Symposium. 2021. pp. 1701–1718.
  13. Косолапов Ю.В. Об упрощении выражений со смешанной битовой и целочисленной арифметикой // Моделирование и анализ информационных систем. 2023. Т. 30. № 2. С. 140–159.
  14. Ceccato M., Tonella P., Basile C., Falcarin P., Torchiano M., Coppens B., De Sutter B. Understanding the behaviour of hackers while performing attack tasks in a professional setting and in a public challenge // Empirical Software Engineering. 2019. no. 24. pp. 240–286.
  15. Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations. Computer Science Technical Reports 148. Department of Computer Science, The University of Auckland, New Zealand. 1997. pp. 1–36.
  16. Mohsen R., Pinto A.M. Evaluating obfuscation security: A quantitative approach // In International Symposium on Foundations and Practice of Security, Springer International Publishing. 2015. pp. 174–192.
  17. Banescu S., Ochoa M., Pretschner A. A framework for measuring software obfuscation resilience against automated attacks // In Proceedings of the 1st International Workshop on Software Protection (SPRO '15). IEEE Press, Piscataway, NJ, USA. 2015. pp. 45–51.
  18. Holder W., McDonald J.T., Anel T.R. Evaluating optimal phase ordering in obfuscation executives // Proceedings of the 7th Software Security, Protection, and Reverse Engineering/Software Security and Protection Workshop. 2017. pp. 1–12.
  19. Collberg C. The Tigress C Diversifier/Obfuscator. 2016. URL: [tigress.cs.arizona.edu/](http://tigress.cs.arizona.edu/) (дата обращения: 23.06.2023).
  20. Kosolapov Y.V., Borisov P.D. Similarity features for the evaluation of obfuscation effectiveness // In 2020 International Conference on Decision Aid Sciences and Application (DASA). 2020. pp. 898–902.
  21. Borisov P.D., Kosolapov Y.V. On the Characteristics of Symbolic Execution in the Problem of Assessing the Quality of Obfuscating Transformations // Aut. Control Comp. Sci. 2022. vol. 56(7). pp. 595–605.
  22. Xiao Y, Guo Y., Wang Y. Metrics for code obfuscation based on symbolic execution and N-scope complexity // Chinese Journal of Network and Information Security. 2022. vol. 8. no. 6. pp. 123–134.
  23. Crescenzo G.D. Cryptographic program obfuscation: Practical solutions and application-driven models // Versatile Cybersecurity. 2018. pp. 141–167.
  24. Gulwani S., Polozov O., Singh R. Program synthesis // Foundations and Trends in Programming Languages. 2017. vol. 4. no. 1-2. pp. 1–119.
  25. Borisov P.D., Kosolapov Y.V. On the Automatic Analysis of the Practical Resistance of Obfuscating Transformations // Aut. Control Comp. Sci. 2020. vol. 54. pp. 619–629.
  26. Walenstein A., El-Ramly M., Cordy J.R., Evans W.S., Mahdavi K., Pizka M., Ramalingam G., von Gudenberg J.W. Similarity in Programs // Duplication, Redundancy, and Similarity in Software, Dagstuhl Seminar Proceedings. 2007. vol. 6301. pp. 1–8.
  27. Ceccato M., Di Penta M., Nagra J., Falcarin P., Ricca F., Torchiano M., Tonella P. The effectiveness of source code obfuscation: An experimental assessment // 17th International Conference on Program Comprehension, IEEE. 2009. pp. 178–187.
  28. Ceccato M., Di Penta M., Falcarin P., Ricca F., Torchiano M., Tonella P. A family of experiments to assess the effectiveness and efficiency of source code obfuscation techniques // Empirical Software Engineering. 2014. vol. 19. pp. 1040–1074.

29. Борисов П.Д., Косолапов Ю.В. Способ оценки похожести программ методами машинного обучения // Труды Института системного программирования РАН. 2022. Т. 34. № 5. С. 63–76.
30. Naville Z. Hikari—an improvement over Obfuscator-LLVM. 2017. URL: <https://github.com/HikariObfuscator/Hikari> (дата доступа: 14.11.2023).
31. Junod P., Rinaldini J., Wehrli J., Michielin J. Obfuscator-LLVM—software protection for the masses // In Proc. of IEEE/ACM 1st International Workshop on Software Protection. 2015. pp. 3–9.
32. Haq I.U., Caballero J. A survey of binary code similarity // ACM Computing Surveys (CSUR). 2021. vol. 54. no. 3. pp. 1–38.
33. Pagani F., Dell’Amico M., Balzarotti D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis // In Proc. of the Eighth ACM Conference on Data and Application Security and Privacy. 2018. pp. 354–365.
34. Ding S.H., Fung B.C., Charland P. Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization // In 2019 IEEE Symposium on Security and Privacy (SP). IEEE. 2019. pp. 472–489.
35. Garba P., Favaro M. Saturn—software deobfuscation framework based on llvm // In Proceedings of the 3rd ACM Workshop on Software Protection. 2019. pp. 27–38.
36. Dinaburg A., Ruef A. Mcsema: Static translation of x86 instructions to LLVM. ReCon 2014 Conference, Montreal, Canada. 2014.
37. Eyrolles N. Obfuscation with Mixed Boolean-Arithmetic Expressions: reconstruction, analysis and simplification tools. Doctoral dissertation. Universite Paris-Saclay, 2017. URL: <https://theses.hal.science/tel-01623849/document> (дата обращения: 14.07.2023).
38. Liang M., Li Z., Zeng Q., Fang Z. Deobfuscation of virtualization-obfuscated code through symbolic execution and compilation optimization // In International Conference on Information and Communications Security. Springer International Publishing, 2018. pp. 313–324.
39. Panchenko M., Auler R., Sakka L., Ottoni G. Lightning BOLT: powerful, fast, and scalable binary optimization // In Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction. 2021. pp. 119–130.
40. Moreira A.A., Ottoni G., Quintao Pereira F.M. Vespa: static profiling for binary optimization // Proceedings of the ACM on Programming Languages. 2021. vol. 5. pp. 1–28.
41. Viticchie A., Regano L., Torchiano M., Basile C., Ceccato M., Tonella P., Tiella R. Assessment of source code obfuscation techniques // 16th international working conference on source code analysis and manipulation (SCAM), IEEE. 2016. pp. 11–20.
42. GCC, the GNU Compiler Collection. URL: <https://gcc.gnu.org/> (дата обращения: 14.07.2023).
43. Clang: a C language family frontend for LLVM. URL: <https://clang.llvm.org/> (дата обращения: 14.07.2023).
44. AMD Optimizing C/C++ and Fortran Compilers (AOCC). URL: <https://developer.amd.com/amd-aocc/> (дата обращения: 14.07.2023).
45. Coreutils – GNU core utilities. URL: <https://www.gnu.org/software/coreutils/> (дата обращения: 14.07.2023).
46. PolyBench/C – the Polyhedral Benchmark suite. URL: <https://web.cse.ohio-state.edu/pouchet.2/software/polybench/> (дата обращения: 14.07.2023).
47. HashCat – advanced password recovery. URL: <https://hashcat.net/hashcat/> (дата обращения: 14.07.2023).

48. small-programs. A set of small programs for experiments with obfuscations. URL: <https://github.com/Boriskin61/small-programs> (дата обращения: 22.07.2023).
49. Куц Д.О. Метод моделирования косвенной адресации в рамках динамической символьной интерпретации. 2023. URL: <https://www.ispras.ru/dcouncil/docs/diss/2023/kuc/dissertacija-kuc.pdf> (дата обращения: 03.09.2023).
50. Лебедев Р.К. Автоматическая генерация хэш-функций для обфускации программного кода // Прикладная дискретная математика. 2020. № 50. С. 102–117.
51. Лебедев В.В. Деобфускация Control Flow Flattening средствами символьного исполнения // Прикладная дискретная математика. Приложение. 2021. № 14. С. 134–138.
52. BinShamlan M.H., Vamatraf M.A., Zain A.A. The impact of control flow obfuscation technique on software protection against human attacks // In 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), IEEE. 2019. pp. 1–5.
53. Xu D. Opaque Predicate: Attack and Defense in Obfuscated Binary Code. Doctoral dissertation, 2018. URL: [https://etda.libraries.psu.edu/files/final\\_submissions/17513](https://etda.libraries.psu.edu/files/final_submissions/17513) (дата обращения: 22.09.2023).
54. Sun Y. Software Protection Algorithm based on Control Flow Obfuscation // International Journal of Performability Engineering. 2018. vol. 14. no. 9. pp. 2181–2188.
55. Kim J., Kang S., Cho E.S., Paik J.Y. LOM: lightweight classifier for obfuscation methods // In Information Security Applications: 22nd International Conference, WISA 2021. Springer International Publishing. 2021. pp. 3–15.
56. Zhao Y., Tang Z., Ye G., Peng D., Fang D., Chen X., Wang Z. Semantics-aware obfuscation scheme prediction for binary. Computers Security. 2020. no. 99. pp. 1–17.
57. Wang C., Hill J., Knight J., Davidson J. Software tamper resistance: Obstructing static analysis of programs. Technical report CS-2000-12. Department of Computer Science, University of Virginia, USA. 2000.
58. Dullien T., Rolles R. Graph-based comparison of executable objects (english version) // Proceedings of the Symposium sur la Securite des Technologies de 'Information et des Communications. 2005. vol. 5. no. 1.

**Борисов Петр Дмитриевич** — заведующий лабораторией, ФГАНУ НИИ "Спецвузавтоматика". Область научных интересов: исследование и анализ программного кода, методы практической обфускации и деобфускации, способы оценки качества обфусцирующих преобразований. Число научных публикаций — 10. [borisovpetr@mail.ru](mailto:borisovpetr@mail.ru); улица Города Волос, 6, 344011, Ростов-на-Дону, Россия; р.т.: +7(863)201-2817.

**Косолапов Юрий Владимирович** — канд. техн. наук, доцент кафедры, кафедра алгебры и дискретной математики, Институт математики, механики и компьютерных наук им. И.И. Воровича Южного федерального университета (ЮФУ). Область научных интересов: помехоустойчивые коды в криптографии и стеганографии, теоретическая и практическая обфускация кода. Число научных публикаций — 100. [uvkosolapov@sfedu.ru](mailto:uvkosolapov@sfedu.ru); улица Мильчакова, 8а, 344090, Ростов-на-Дону, Россия; р.т.: +7(863)297-5111.

P. BORISOV, Yu. KOSOLAPOV  
**A METHOD TO QUANTITATIVE COMPARE OBFUSCATING  
TTRANSFORMATIONS**

*Borisov P., Kosolapov Yu. A Method to Quantitative Compare Obfuscating Ttransformations.*

**Abstract.** The paper considers the problem of quantitative comparison of potency and resistance of practically applied obfuscating transformations of program code. A method is proposed to find the potency and resistance of transformations by calculating the "comprehensibility" of the obfuscated and deobfuscated versions of a program, respectively. As a measure of program comprehensibility, it is proposed to use the similarity of this program to the approximation of its "most comprehensible" version. Based on the proposed method a model to assess potency and resistance was built, the main elements of which are: a set of investigated obfuscating transformations, a similarity function, a method to approximate the most comprehensible version of the program and a deobfuscator. To implement this model 1) obfuscating transformations provided by Hikari obfuscator are chosen; 2) 8 similarity functions are constructed by machine learning methods using static characteristics of programs from CoreUtils, PolyBench and HashCat sets; 3) the smallest program version was chosen as an approximation of the most comprehensible program version (found among the versions obtained using optimization options of GCC, Clang and AOCC compilers); 4) a program deobfuscation scheme based on the optimizing compiler from LLVM was built and implemented. The results of the potency and resistance for sequences of transformations of lengths one, two and three were experimentally obtained. These results showed consistency with the results of independent potency and resistance evaluations obtained by other methods. In particular, it was found that the highest potency and resistance are demonstrated by sequences of transformations starting with transformations of the control flow graph, and the lowest resistance and potency are generally demonstrated by sequences that do not contain such transformations.

**Keywords:** obfuscation, executable code, efficiency, resistance, similarity.

## References

1. Akhuzada A., Sookhak M, Anuar A.B., Gani A., Ahmed E., Shiraz M., Furnell S., Hayat A., Khan M.K. Man-At-The-End attacks: Analysis, taxonomy, human aspects, motivation and future directions. *Journal of Network and Computer Applications*. 2015. vol. 48. pp. 44–57.
2. Undrits R., Resende J. et. al. *CyberSec4Europe D3.23: Cybersecurity Outlook 2. Research Report D3.23*. 2022. pp. 1–82.
3. Biernacki L., Gallagher M., Xu Z., Aga M.T., Harris A., Wei S., Tiwari M., Kasikci B., Malik S., Austin T. Software-driven security attacks: From vulnerability sources to durable hardware defenses. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*. 2021. vol. 17. no. 3. pp. 1–38.
4. Varnovsky N.P., Zakharov V.A., Kuzurin N.N., Shokurov V.A. [The current state of art in program obfuscations: definitions of obfuscation security]. *Trudy Instituta sistemnogo programirovaniya RAN – Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2014. vol. 26. no. 3 pp. 167–198. (In Russ.).
5. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vadhan S., Yang K. On the (im) possibility of obfuscating programs. *Journal of the ACM (JACM)*. 2012. vol. 59. no. 2. pp. 1–42.

6. Zobernig L. *Mathematical Aspects of Program Obfuscation*. Doctoral dissertation. ResearchSpace@ Auckland, 2020. Available at: [www.math.auckland.ac.nz/~sgal018/Lukas-Zobernig-Thesis.pdf](http://www.math.auckland.ac.nz/~sgal018/Lukas-Zobernig-Thesis.pdf) (accessed 19.06.2023).
7. Garg S., Gentry C., Halevi S., Raykova M., Sahai A., Waters B. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*. 2016. vol. 45. no. 3. pp. 882–929.
8. Kochberger P., Schrittwieser S., Coppens B., De Sutter B. Evaluation Methodologies in Software Protection Research. *arXiv preprint arXiv:2307.07300*. 2023. pp. 1–67.
9. Zhou Y., Main A., Gu Y.X., Johnson H. Information hiding in software with mixed boolean-arithmetic transforms. *Information Security Applications: 8th International Workshop on Information Security Applications*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007. vol. 4867. pp. 61–75.
10. Reichenwallner B., Meerwald-Stadler P. Efficient Deobfuscation of Linear Mixed Boolean-Arithmetic Expressions. *Proceedings of the 2022 ACM Workshop on Research on offensive and defensive techniques in the context of Man At The End (MATE) attacks*. 2022. pp. 19–28.
11. Xu D., Liu D., Feng W., Ming J., Zheng Q., Li J., Yu Q. Boosting SMT solver performance on mixed-bitwise-arithmetic expressions. *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*. 2021. pp. 651–664.
12. Liu B., Shen J., Ming J., Zheng Q., Li J., Xu D. MBA-Blast: Unveiling and Simplifying Mixed Boolean-Arithmetic Obfuscation. *Proceedings of the 30th USENIX Security Symposium*. 2021. pp. 1701–1718.
13. Kosolapov Yu.V. [On Simplifying Expressions with Mixed Boolean-Arithmetic]. *Modelirovanie i analiz informacionnyh sistem – Modeling and Analysis of Information Systems*. 2023. vol. 30. no. 2. pp. 140–159. (In Russ.).
14. Ceccato M., Tonella P., Basile C., Falcarin P., Torchiano M., Coppens B., De Sutter B. Understanding the behaviour of hackers while performing attack tasks in a professional setting and in a public challenge. *Empirical Software Engineering*. 2019. no. 24. pp. 240–286.
15. Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations. *Computer Science Technical Reports 148*. Department of Computer Science, The University of Auckland, New Zealand. 1997. pp. 1–36.
16. Mohsen R., Pinto A.M. Evaluating obfuscation security: A quantitative approach. In *International Symposium on Foundations and Practice of Security*, Springer International Publishing, 2015. pp. 174–192.
17. Banescu S., Ochoa M., Pretschner A. A framework for measuring software obfuscation resilience against automated attacks. In *Proceedings of the 1st International Workshop on Software Protection (SPRO '15)*. IEEE Press, Piscataway, NJ, USA. 2015. pp. 45–51.
18. Holder W., McDonald J.T., Andel T.R. Evaluating optimal phase ordering in obfuscation executives. *Proceedings of the 7th Software Security, Protection, and Reverse Engineering/Software Security and Protection Workshop*. 2017. pp. 1–12.
19. Collberg C. *The Tigress C Diversifier/Obfuscator*. 2016. Available at: [tigress.cs.arizona.edu/](http://tigress.cs.arizona.edu/) (accessed 23.06.2023).
20. Kosolapov Y.V., Borisov P.D. Similarity features for the evaluation of obfuscation effectiveness. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*. 2020. pp. 898–902.
21. Borisov P.D., Kosolapov Y.V. On the Characteristics of Symbolic Execution in the Problem of Assessing the Quality of Obfuscating Transformations. *Aut. Control Comp. Sci.* 2022. vol. 56(7). pp. 595–605.

22. Xiao Y, Guo Y., Wang Y. Metrics for code obfuscation based on symbolic execution and N-scope complexity. *Chinese Journal of Network and Information Security*. 2022. vol. 8. no. 6. pp. 123–134.
23. Crescenzo G.D. Cryptographic program obfuscation: Practical solutions and application-driven models. *Versatile Cybersecurity*. 2018. pp. 141–167.
24. Gulwani S., Polozov O., Singh R. Program synthesis. *Foundations and Trends in Programming Languages*. 2017. vol. 4. no. 1-2. pp. 1–119.
25. Borisov P.D., Kosolapov Y.V. On the Automatic Analysis of the Practical Resistance of Obfuscating Transformations. *Aut. Control Comp. Sci*. 2020. vol. 54. pp. 619–629.
26. Walenstein A., El-Ramly M., Cordy J.R., Evans W.S, Mahdavi K., Pizka M., Ramalingam G., von Gudenberg J.W. Similarity in Programs. Duplication, Redundancy, and Similarity in Software, Dagstuhl Seminar Proceedings. 2007. vol. 6301. pp. 1–8.
27. Ceccato M., Di Penta M., Nagra J., Falcarin P., Ricca F., Torchiano M., Tonella P. The effectiveness of source code obfuscation: An experimental assessment. *17th International Conference on Program Comprehension, IEEE*. 2009. pp. 178–187.
28. Ceccato M., Di Penta M., Falcarin P., Ricca F., Torchiano M., Tonella P. A family of experiments to assess the effectiveness and efficiency of source code obfuscation techniques. *Empirical Software Engineering*. 2014. vol. 19. pp. 1040–1074.
29. Borisov P.D., Kosolapov Yu.V. [Method to Evaluate Program Similarity Using Machine Learning Methods]. *Trudy Instituta sistemnogo programirovaniya RAN – Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2023. vol. 34. no. 5. pp. 63–76. (In Russ.).
30. Naville Z. Hikari—an improvement over Obfuscator-LLVM. 2017. Available at: <https://github.com/HikariObfuscator/Hikari> (accessed 14.11.2023).
31. Junod P., Rinaldini J., Wehrli J., Michielin J. Obfuscator-LLVM—software protection for the masses. In *Proc. of IEEE/ACM 1st International Workshop on Software Protection*. 2015. pp. 3–9.
32. Haq I.U., Caballero J. A survey of binary code similarity. *ACM Computing Surveys (CSUR)*. 2021. vol. 54. no. 3. pp. 1–38.
33. Pagani F., Dell’Amico M., Balzarotti D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis. In *Proc. of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018. pp. 354–365.
34. Ding S.H., Fung B.C., Charland P. Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019. pp. 472–489.
35. Garba P., Favaro M. Saturn-software deobfuscation framework based on llvm. In *Proceedings of the 3rd ACM Workshop on Software Protection*. 2019. pp. 27–38.
36. Dinaburg A., Ruef A. Mcsema: Static translation of x86 instructions to LLVM. *ReCon 2014 Conference*, Montreal, Canada. 2014.
37. Eyrolles N. Obfuscation with Mixed Boolean-Arithmetic Expressions: reconstruction, analysis and simplification tools. *Doctoral dissertation*. Universite Paris-Saclay, 2017. Available at: <https://theses.hal.science/tel-01623849/document> (accessed 14.07.2023).
38. Liang M., Li Z., Zeng Q., Fang Z. Deobfuscation of virtualization-obfuscated code through symbolic execution and compilation optimization. In *International Conference on Information and Communications Security*. Springer International Publishing, 2018. pp. 313–324.
39. Panchenko M., Auler R., Sakka L., Ottoni G. Lightning BOLT: powerful, fast, and scalable binary optimization. In *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction*. 2021. pp. 119–130.

40. Moreira A.A., Ottoni G., Quintao Pereira F.M. Vespa: static profiling for binary optimization. *Proceedings of the ACM on Programming Languages*. 2021. vol. 5. pp. 1–28.
41. Viticchie A., Regano L., Torchiano M., Basile C., Ceccato M., Tonella P., Tiella R. Assessment of source code obfuscation techniques. 16th international working conference on source code analysis and manipulation (SCAM), IEEE. 2016. pp. 11–20.
42. GCC, the GNU Compiler Collection. Available at: <https://gcc.gnu.org/> (accessed 14.07.2023).
43. Clang: a C language family frontend for LLVM. Available at: <https://clang.llvm.org/> (accessed 14.07.2023).
44. AMD Optimizing C/C++ and Fortran Compilers (AOCC). Available at: <https://developer.amd.com/amd-aocc/> (accessed 14.07.2023).
45. Coreutils – GNU core utilities. Available at: <https://www.gnu.org/software/coreutils/> (accessed 14.07.2023).
46. PolyBench/C – the Polyhedral Benchmark suite. Available at: <https://web.cse.ohio-state.edu/pouchet.2/software/polybench/> (accessed 14.07.2023).
47. HashCat – advanced password recovery. Available at: <https://hashcat.net/hashcat/> (accessed 14.07.2023).
48. small-programs. A set of small programs for experiments with obfuscations. Available at: <https://github.com/Boriskin61/small-programs> (accessed 22.07.2023).
49. Kutz D.O. Method for modeling indirect addressing within dynamic symbolic interpretation. Doctorial dissertation, Moscow, 2023. Available at: <https://www.ispras.ru/dcouncil/docs/diss/2023/kuc/dissertacija-kuc.pdf> (accessed 03.09.2023).
50. Lebedev R.K. [Automatic generation of hash functions for program code obfuscation]. *Prikladnaya Diskretnaya Matematika – Applied Discrete Mathematics*. 2020. no. 50. pp. 102–117. (In Russ.).
51. Lebedev V.V. [Control Flow Flattening deobfuscation using symbolic execution]. *Prikladnaya Diskretnaya Matematika – Applied Discrete Mathematics*. 2021. no. 14. pp. 134–138. (In Russ.).
52. BinShamlan M.H., Bamatraf M.A., Zain A.A. The impact of control flow obfuscation technique on software protection against human attacks. In 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), IEEE. 2019. pp. 1–5.
53. Xu D. Opaque Predicate: Attack and Defense in Obfuscated Binary Code. Doctoral dissertation, 2018. Available at: [https://etda.libraries.psu.edu/files/final\\_submissions/17513](https://etda.libraries.psu.edu/files/final_submissions/17513) (accessed 22.09.2023).
54. Sun Y. Software Protection Algorithm based on Control Flow Obfuscation. *International Journal of Performability Engineering*. 2018. vol. 14. no. 9. pp. 2181–2188.
55. Kim J., Kang S., Cho E.S., Paik J.Y. LOM: lightweight classifier for obfuscation methods. In *Information Security Applications: 22nd International Conference, WISA 2021*. Springer International Publishing. 2021. pp. 3–15.
56. Zhao Y., Tang Z., Ye G., Peng D., Fang D., Chen X., Wang Z. Semantics-aware obfuscation scheme prediction for binary. *Computers Security*. 2020. no. 99. pp. 1–17.
57. Wang C., Hill J., Knight J., Davidson J. Software tamper resistance: Obstructing static analysis of programs. Technical report CS-2000-12. Department of Computer Science, University of Virginia, USA. 2000.
58. Dullien T., Rolles R. Graph-based comparison of executable objects (english version). *Proceedings of the Symposium sur la Securite des Technologies de'Information et des Communications*. 2005. vol. 5. no. 1.



**Borisov Petr** — Head of the laboratory, FSASE SRI "Specvuzavtomatika". Research interests: program code analysis, obfuscation and deobfuscation, methods for assessing the quality of obfuscating transformations. The number of publications — 10. borisovpetr@mail.ru; 6, Goroda of Volos St., 344011, Rostov-on-Don, Russia; office phone: +7(863)201-2817.

**Kosolapov Yury** — Ph.D., Associate professor of the department, Department of algebra and discrete mathematics, I.I. Vorovich Institute of Mathematics, Mechanics and Computer Sciences of the Southern Federal University (SFedU). Research interests: error-correcting codes in cryptography and steganography, theoretical and practical code obfuscation. The number of publications — 100. yvkosolapov@sfedu.ru; 8a, Milchakova St., 344090, Rostov-on-Don, Russia; office phone: +7(863)297-5111.

А.Н. Ивутин, А.С. Новиков, М.С. Пестин, А.Г. Волошко  
**ДЕЦЕНТРАЛИЗОВАННЫЙ ПРОТОКОЛ ОРГАНИЗАЦИИ  
УСТОЙЧИВОГО ВЗАИМОДЕЙСТВИЯ АБОНЕНТОВ В СЕТЯХ  
С ВЫСОКОЙ ДИНАМИКОЙ ИЗМЕНЕНИЯ ТОПОЛОГИИ**

*Ивутин А.Н., Новиков А.С., Пестин М.С., Волошко А.Г. Децентрализованный протокол организации устойчивого взаимодействия абонентов в сетях с высокой динамикой изменения топологии.*

**Аннотация.** Аварийным службам часто приходится проводить спасательные и ликвидационные работы в условиях отсутствия централизованной связи. Невозможность обеспечения стабильной коммуникации между членами спасательного подразделения существенно снижает качество проведения работ. При этом в современных реалиях под стабильной коммуникацией понимается не только голосовой обмен, который может быть обеспечен коротковолновыми радиопередатчиками, но и интенсивный обмен большими объемами трафика. Применение стандартных решений на основе типового сетевого оборудования (Wi-Fi, спутниковая связь и др.) и существующих алгоритмов обеспечения качества обслуживания в рассматриваемых условиях не позволяет быстро обеспечить информационный обмен между разнородными абонентами. Более того, работа в высокочастотных диапазонах может быть сильно затруднена при наличии препятствий, что снижает общую площадь покрытия и качество передачи данных. Мы предлагаем протокол маршрутизации сетевого уровня, предназначенный для организации децентрализованной связи в подразделении аварийной службы, где абоненты отличаются разной степенью мобильности и типом передаваемого трафика. Данный протокол включает алгоритмы подключения к сети, обнаружения оптимального и альтернативных маршрутов связи, передачи и балансировки трафика по найденным маршрутам. Оригинальный алгоритм поиска маршрутов анализирует производительность каналов связи и определяет все возможные пути передачи трафика между абонентами. С использованием функции оценки маршрутов, основанной на градиентном бустинге деревьев принятия решений, производится формирование оптимальных и альтернативных маршрутов связи, а при передаче данных, на основе полученной информации, выполняется балансировка трафика. Экспериментальное исследование предложенного протокола показало улучшение показателей скорости развертывания и качества обслуживания на сценариях с различной степенью мобильности абонентов.

**Ключевые слова:** беспроводная децентрализованная сеть, протокол маршрутизации, метрики маршрутизации, градиентный бустинг, коэффициент доставки пакетов, сквозная задержка, качество обслуживания.

**1. Введение.** Успех проведения операций по ликвидации чрезвычайных ситуаций зависит от наличия эффективной и надёжной связи между её участниками на территориях от десятков до сотен квадратных километров. Спектр участников весьма разнообразен (рисунок 1). Во-первых, это могут быть группы спасателей, экипажи специальной техники (автомобили общего назначения, мобильные лаборатории, авиатехника и другие), пункты управления операциями, где в качестве абонентов выступают люди. Во-вторых, ими являются

автономные и радиоуправляемые мобильные роботы (наземные, водные или летательные), стационарные системы мониторинга. Это обуславливает потребность в передачи разнородного трафика, в том числе аудио- и видеопотоков, телеметрии от датчиков и систем мониторинга, команд управления, текстовых сообщений.

Применение централизованных инфраструктурных сетей ограничено в условиях чрезвычайной ситуации. Во-первых, необходимая инфраструктура может отсутствовать в районе аварии или она может быть разрушена в результате стихийного бедствия, ставшего причиной ЧП (землетрясения, ураганы, пожары, наводнения). Развёртывание инфраструктурной сети, обеспечивающей связь на значительные расстояния – трудоёмкая задача, требующая высоких временных и материальных затрат. Это недопустимо в условиях проведения ликвидационных и спасательных работ.



Рис. 1. Узлы MANET в аварийно-спасательной службе

Беспроводная коротковолновая радиосвязь [1] способна обеспечивать покрытие на сотни километров в условиях сильнопересечённой местности. Однако, низкая скорость передачи данных (менее 6 килобит в секунду) существенно ограничивает возможности трансляции цифровых данных. Поэтому применение КВ-радиосвязи ограничивается, в основном, передачей голосовых сообщений.

Современная спутниковая связь [2, 3] может обеспечивать высокую скорость обмена данными на приём при существенно меньших значениях на передачу. При этом используемое оборудование имеет относительно большие габариты, высокий расход электроэнергии, и требует соблюдения направленности антенны на спутник [4]. Особенно остро эти проблемы проявляются, когда абоненты – подвижные узлы.

Одним из вариантов решения проблемы является развёртывание точки доступа Wi-Fi [5] на основе стандартов IEEE 802.11 [6]. Однако рабочий диапазон радиоволн (ELF и SLF [7]) не позволяет эффективно использовать технологию на сильно пересеченной местности на больших расстояниях.

Альтернативным подходом является построение мобильных беспроводных ad-hoc сетей (mobile ad-hoc network, MANET), использующих ячеистую топологию (mesh) [8, 9]. Каждый узел является оконечной точкой, и в то же время, ретранслятором для других узлов. Данные сети устойчивы к изменению топологии. Маршруты передачи данных формируются уже непосредственно во время работы сети и со временем корректируются или повторно определяются. Такие сети не требуют специализированного аппаратного обеспечения и могут быть построены с использованием стандартного оборудования, как УКВ-радио, Wi-Fi - передатчики и др.

Одной из ключевых технологий MANET являются специальные протоколы маршрутизации. Их задачами является построение маршрутов связи между удалёнными абонентами, поддержание маршрутов связи в актуальном состоянии, исправление неисправностей в маршрутах, обеспечение ретрансляции данных. Протоколы маршрутизации разделяются на проактивные, реактивные, гибридные и географические.

Проактивными (или табличными) называются такие протоколы маршрутизации, в которых сетевые узлы производят периодический обмен служебных сообщений с информацией об известных связях между абонентами, на основании чего производится формирования таблиц маршрутизации. Наиболее распространёнными протоколами текущей категории являются DSDV, OLSR и решения на их основе [10]. Однако постоянный служебный трафик, требуемый для поддержания таблиц маршрутизации, снижению пропускной способности даже в условиях простаивания сети, когда полезные данные передаются в малом объёме. Эти ограничивает применение табличных протоколов маршрутизации в высокодинамичных сетях передачи данных.

Реактивные (или «по запросу») протоколы маршрутизации производят поиск маршрута к целевому узлу только при необходимости. В режиме простоя сети или при отсутствии необходимости передачи данных между абонентами обнаружение маршрутов производиться не будет. Классические протоколами маршрутизации «по запросу» – AODV, DYMO, DSR, TORA [11]. Реактивные протоколы маршрутизации лучше проактивных адаптированы под функционирование высокодинамичных ad-hoc сетей, однако они обладают сопутствующими недостатками (высокой задержкой на обнаружение маршрутов, возникновение «всплесков» служебного трафика во время поиска маршрутов, высокие накладные расходы на восстановление маршрутов). Кроме того, имеются существенные затраты на повторный поиск маршрутов связи.

Существующие гибридные протоколы маршрутизации (ZRP, FSR, LANMAR [12]) сочетают в себя свойства реактивных и проактивных протоколов маршрутизации. Их особенностью является использование проактивной маршрутизации к ближайшим абонентам, а к более дальним – реактивной. Однако эффективность такой схемы существенно снижается в условиях различия в мобильности абонентов.

Также улучшением стандартных подходов являются схемы маршрутизации с позиционированием на основе глобальных навигационных систем (GPS, Глонасс), где при построении маршрутов учитывается взаимное положение абонентов в пространстве (протоколы GAF, GPSR, LAR [13]). Потребность в специальном оборудовании, потребность во взаимном оповещении абонентов сети об изменении своего местоположения, отсутствие учёта препятствий и помех делает подход узкоспециализированным.

Таким образом, имеющиеся решения эффективно функционируют в сетях со стабильной или квазистабильной топологией. Существующие подходы не обеспечивают эффективного функционирования в сетях с высокой динамикой изменения топологии (из-за высокой мобильности узлов, наличия множества помех). При использовании одного канала связи между абонентами пропускная способность ограничивается из-за перегрузок. Это требует обнаружения и использования нескольких каналов связи между абонентами. Большинство классических подходов маршрутизации трафика в MANET не предполагают механизмов балансировки трафика. При этом балансировка трафика возможна в том случае, если поисковые алгоритмы способны строить несколько маршрутов к узлу

назначения, при этом маршруты должны быть, по большей части, непересекающимися.

Одним из подходов к повышению производительности MANET является использование протоколов многопутевой маршрутизации. Они имеют ряд преимуществ, таких как возможность резервирования части маршрутов на случай недоступности основного канала и обеспечения параллельной передачи трафика по нескольким маршрутам одновременно.

Таким образом, предлагается решение, предоставляющее возможность организации устойчивой связи в условиях нестабильной топологии сети и низкой пропускной способности каналов связи за счёт обеспечения параллельных потоков передачи данных и применение оригинальных методов балансировки трафика для отдельных абонентов. Предлагаемое решение обеспечивает развёртывание распределённой системы связи за минимальное время, что особенно актуально для работы поисково-спасательных подразделений при чрезвычайных ситуациях, когда стандартные технологии организации связи не функционируют.

В разделе 2 мы описываем предыдущее и связанные работы по проблемам многопутевой маршрутизации данных и балансировки трафика по сети. В разделе 3 мы представляем протокол маршрутизации. Здесь рассматриваются общая структура протокола маршрутизации, используемые им структуры данных, описание и оценка маршрутов связи, основные алгоритмы функционирования протокола. В разделе 4 мы производим экспериментальное исследование, в ходе которого сравниваем показатели производительности разработанного протокола со стандартным AOMDV. В заключении мы подводим итоги нашей работы.

**2. Обзор литературы по теме исследования.** В данном разделе рассматриваются решения многопутевой маршрутизации, предназначенные для обеспечения высоких показателей качества обслуживания (quality of service, QoS) в MANET. Под QoS в MANET понимается способность сети обеспечить необходимый сервис заданному трафику в определенных технологических рамках [14].

Многопутевая версия протокола AODV носит название AOMDV (ad-hoc on-demand multipath distance vector routing) [15]. AOMDV позволяет обнаруживать несколько независимых маршрутов к узлу назначения. Альтернативные маршруты используются, когда основной маршрут становится неисправным. Оптимизированный AOMDV (optimized AOMDV, OAOMDV) решает проблему «отсечения маршрута» [16]. Предложенная схема предполагает следующее. Узел,

из которого выходит несколько восходящих маршрутов, возвращает ответы RREP АСК (подтверждение маршрута) всем нисходящим маршрутам. Недостатком схемы является то, что при большом количестве нисходящих маршрутов в сети образуется много пакетов RREP АСК. По грубым оценкам он может занимать до 20% служебного трафика.

В ряде работ для обеспечения высоких показателей качества функционирования беспроводной сети используются интеллектуальные подходы. В работе [17] для обеспечения многопутевой маршрутизации используются алгоритм выбора биологического аттрактора. Подход предназначен для повышения коэффициента доставки пакетов и средней пропускной способности. В статье [18] предлагается алгоритм маршрутизации, основанный на оптимизации методом кукушки (cuckoo optimization algorithm, COA). Протокол прогнозируемой энергоэффективной и надежной многоадресной маршрутизации (predictive energy efficient and reliable multicast routing in MANET, PEERMR) [19] использует алгоритм PSO для построения надежного, энергоэффективного многоадресного дерева маршрутов. Еще одной схемой обнаружения многоадресных маршрутов для MANET является стратегия выбора маршрута под названием «Энергоэффективная многоадресная рассылка с учетом всего срока службы» (energy-efficient lifetime aware multicast, EELAM) [20], которая основана на применении адаптивного генетического алгоритма. EELAM использует адаптивную эволюционную стратегию для построения древовидной топологии маршрутов в сети с минимальным потреблением энергии при передаче трафика. В статье [21] предложен механизм маршрутизации SynFAnt, основанный на нечёткой синхронизированной сети Петри (SynFPN) для моделирования функций маршрутизации и принятия решений, а также алгоритме муравьиной колонии для поиска решения проблемы событий неопределенности в ad-hoc сетях.

В статье [22] предлагается протокол маршрутизации LCMR (least common multiple based routing), который предполагает построение нескольких маршрутов связи и балансировку трафика между ними. Пакеты трафика распределяются обратно пропорционально времени, затраченного на обнаружения маршрута. Во всех вышеприведённых подходах программные службы не взаимодействуют между собой для корректировки весов маршрутов, поэтому подход применим исключительно для сетей со стабильными параметрами работы.

Ряд решений на основе AODMV предлагают использование альтернативных метрик для обеспечения QoS. Протокол QoS-AODMV использует маршрутную метрику на основе длины и остаточной энергии [23]. Другие решения на основе AODMV используют маршрутные метрики на основе остаточной энергии абонентов, а также входных мощностей сигналов [24 – 28].

Протокол LRMR (link reliable multipath routing) из работы [29] определяет несколько параллельных маршрутов на основе длины пути и оценки качества маршрута. При этом не учитывается загруженность абонентов и каналов связи, остаточная энергия сетевых узлов. В других работах также предлагается учитывать качество каналов связи. Авторы [30] предлагают в своём протоколе для измерения качества канала использовать время отклика соседнего узла на отправленное сообщение. Авторы работ [31, 32] в качестве метрики надёжности в алгоритме поиска маршрута предлагают использовать уровень входящего сигнала. Недостатками работ является невозможность оценки стабильности канала связи при высоком уровне мобильности абонентов. В статье С. Амхамди [33] рассматривается протокол маршрутизации с балансировкой трафика, целью которого распределение энергетической нагрузки на узлы сети. В качестве метрики рассматривается энергопотребления абонентами маршрутов. В работе [34] производится балансировка трафика по найденным маршрутам на основе показателей средней очереди на узлах маршрута.

Предложенные подходы могут обеспечивать QoS в частных случаях и по отдельным показателям, однако они не способны обеспечивать высокую производительность ввиду того, что учитываются лишь отдельные частные маршрутные метрики, но не их совокупность.

Рассмотренные методы многопутевой маршрутизации трафика имеют недостатки. Во многих подходах параллельные маршруты исключительно резервируются на будущее. Трафик по ним не передаётся, из-за чего они простаивают. В существующих методах балансировка применяется для обеспечения производительности сети не по интегральному, а по отдельному критерию, что не позволяет учесть всех особенностей высокодинамичной MANET сети. При этом принято использовать исключительно не пересекающиеся маршруты, что в реальных сетях достаточно трудно обеспечить ввиду проблем отсечения маршрута через узел или канал.

В соответствии с этим, разрабатываемый протокол должен соответствовать следующим требованиям:



- функционировать по реактивным алгоритмам обнаружения маршрута, использовать лучшие стороны протоколов маршрутизации по запросу;
- быть децентрализованным. Это означает, что узлы не должны иметь жестко закрепленных за собой функций, являясь одновременно оконечной точкой и маршрутизатором;
- обеспечивать минимальное количество потерь пакетов данных при передаче по маршрутам связи при сохранении приемлемых значений—других показателей качества обслуживания, в частности время доставки данных, пропускную способность сети;
- обеспечивать эффективную маршрутизацию трафика для абонентов с разной мобильностью. В качестве критериев эффективности могут выступать доля доставленного трафика по маршруту, время доставки пакетов данных;
- обеспечивать распределённую балансировку трафика при передаче данных для оптимизации показателей качества обслуживания, а также оперативно обновлять данные по маршрутам связи, задействованным в балансировке трафика по сети.

### **3. Протокол организации связи и передачи данных.**

Протокол маршрутизации должен обеспечивать установление соединения с абонентами, допускающие непосредственную связь друг с другом без промежуточных ретрансляторов (соседние абоненты), обнаружение маршрутов, маршрутизацию трафика по найденным маршрутам. Технология MANET подразумевает отсутствие связи «каждый с каждым» в топологии сети. В результате полная связанность может быть достигнута через произвольное (достаточно большое) число промежуточных узлов-ретрансляторов (рисунок 6).

Разработанный протокол представляет совокупность нескольких алгоритмов, при помощи которых организуется связь между абонентами:

- установление абонентом соединения с сетью;
- поиск маршрутов связи к целевому абоненту;
- повторный поиск маршрутов связи при их отсутствии;
- передача данных и балансировка трафика.

Также протокол включает в себя структуры, при помощи которых производится хранение данных о соединениях с абонентами и маршрутами связи до них, функции оценки маршрутов связи.

**3.1. Хранение маршрутов связи.** Для хранения информации о сетевых агентах, наличия связи с соседними абонентами и маршрутов передачи трафика мы будем использовать ассоциативный массив AgentMap (рисунок 2). Ключом является адрес абонента A. Значение –

структура AgentInformation. AgentInformation содержит следующие поля:

- Direct – флаг наличия прямой связи с абонентом A;
- LastTime – время, когда в последний раз было получено NeighborMessage или NeighborMessageAnswer от абонента A;
- LastRouteID – последний использованный идентификатор поиска маршрута к абоненту A;
- NeighborList – список смежных абонентов для A (актуально, если Direct = True);
- RouteMap – ассоциативный массив, содержащий информацию о наличии маршрутов к A, пролегающих через своего абонента.

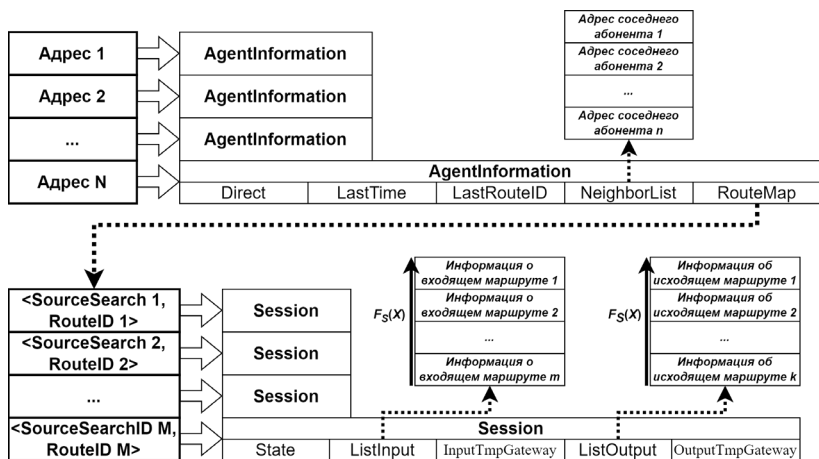


Рис. 2. Структура AgentMap

Ключом RouteMap является пара значений  $\langle \text{SourceSearch}, \text{RouteID} \rangle$ . SourceSearch – адрес абонента, который инициировал поиск к абоненту A. RouteID – идентификатор поиска маршрутов. Если SourceSearch = SelfA, то текущий абонент является инициатором поиска маршрута к A. Каждой паре  $\langle \text{SourceSearch}, \text{RouteID} \rangle$  соответствует сессия поиска маршрутов. Сессия представляет временной промежуток от начала поиска маршрутов к A до момента их устаревания. Структура Session имеет следующие поля:

- State – состояние сессии. Всего имеется 3 состояния: запущена процедура поиска маршрута, имеются актуальные маршруты, сессия устарела;

- ListInput – список входящих маршрутов к A;
- ListOutput – список исходящих маршрутов к A;
- InputTmpGateway – адрес абонента, который предшествует текущему абоненту во временном маршруте;
- OutputTmpGateway – адрес абонента, который следует за текущим абонентом во временном маршруте.

Структура AgentMap формируется на этапах установления соединения с сетью и обнаружения маршрутов к абонентам. Полученная информация используется для выбора путей передачи трафика к целевому абоненту.

**3.2. Установление соединения с сетью.** Установить соединение с сетью в MANET – означает установить прямую связь с несколькими соседними абонентами. При этом имеются соседи первого и второго порядков. Соседи первого порядка – напрямую доступны для текущего абонента. Соседи второго порядка доступны транзитивно через соседей первого порядка.

Для обмена трафиком в рамках процедуры установления и поддержания соединения с сетью были разработаны пакеты NeighborMessage и NeighborMessageAnswer (рисунок 3). NeighborMessage используется для уведомления других абонентов о своём присутствии в сети. NeighborMessageAnswer используется для подтверждения получения NeighborMessage соседними абонентами первого порядка.

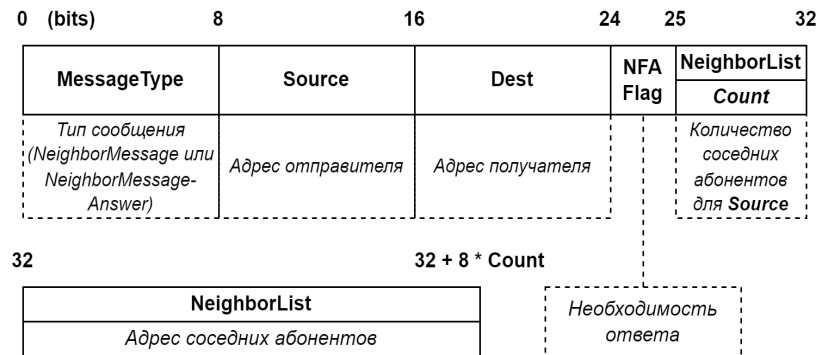


Рис. 3. Структура NeighborMessage и NeighborMessageAnswer

Каждый HelloInterval производится широковещательная рассылка пакета NeighborMessage. Он включает актуальный на момент отправки список соседних абонентов первого порядка. Опционально

может быть затребовано подтверждение получения пакета установкой флага NFAFlag. В этом случае принявший NeighborMessage абонент должен отправить ответ при помощи сообщения NeighborMessageAnswer, структура которого аналогична NeighborMessage. Связь с соседним абонентом считается установленной, если в течение каждого периода  $2 * \text{HelloInterval}$  от него поступает сообщение NeighborMessage или NeighborMessageAnswer. В ином случае абонент удаляется из списка соседних узлов. Схема алгоритма для подключения к сети и поддержания связи с соседними абонентами представлена на рисунке 4.

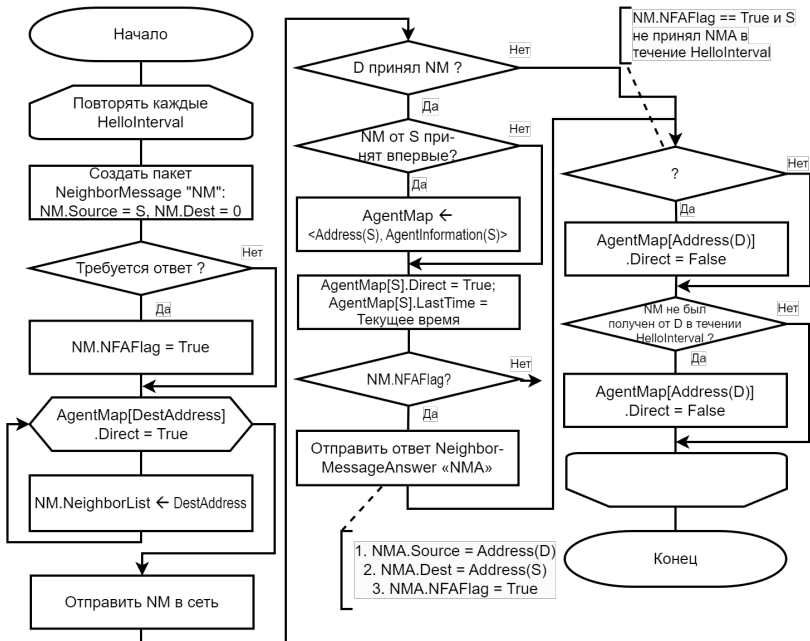


Рис. 4. Схема алгоритма подключения узла к сети и поддержания связи с соседними абонентами

Таким образом, при выполнении алгоритма Hello производится формирование таблицы TableAbon, которая задействуется в дальнейшем в процессе поиска маршрутов между узлами.

**3.3. Описание маршрутов передачи данных.** Производить оценку маршрутов связи можно на основе множества частных показателей, характеризующих функционирование абонентов

и каналов связи между ними. Эффективность маршрутизации зависит от того, насколько каждый абонент или канал связи эффективно продвигают пакет в процессе передачи данных.

В качестве признаков описания маршрута использовались следующие:

- $N$  – длина маршрута;
- $V_{av}$ ,  $V_{max}$  – средняя и максимальная скорости перемещения абонентов внутри маршрута;
- $Dist_{av}$ ,  $Dist_{max}$  – среднее и максимальное расстояние между узлами внутри маршрута;
- $I_{all}$ ,  $I_{max}$  – общее и максимальное внутри узла количество пересечений маршрутов между другими узлами;
- $L_{av}$ ,  $L_{max}$  – средняя и максимальная загрузки каналов связи;
- $Q_{av}$ ,  $Q_{max}$  – средний и максимальный размеры очереди пакетов на транзитных абонентах;
- $Delay_{av}$ ,  $Delay_{max}$  – средняя и максимальная задержки пакета на узлах маршрута;
- $R_{ch}$  – доля данных, которые не теряются при прохождении пакетов через каналы беспроводной связи;
- $R_n$  – доля данных, которые не теряются при прохождении пакетов через абонентов сети.

Совокупность метрик эффективности функционирования маршрутов образует его признаковое описание  $X$ . Используя его, можно выдать оценку о степени его применимости для маршрутизации трафика. За счёт того, что при построении маршрутов учитываются показатели загрузки каналов и абонентов связи, количество пересекающихся актуальных маршрутов, пропускную способность сети, будет обеспечиваться высокое QoS при передаче данных на данном маршруте.

**3.4. Оценка маршрутов связи.** Для оценки маршрутов в предлагаемых нами алгоритмах связи введём функцию оценки  $F_S(X)$ .  $F_S(X)$  будет использоваться для оценки пригодности частично или полностью сформированного маршрута для передачи трафика по сети. Основные показатели, которые характеризуют качество передачи данных по маршруту связи – его надёжность и оперативность доставки данных. Надёжность может быть выражена долей доставленных пакетов (packed delivered ratio, PDR). Оперативность – сквозной задержкой передачи данных (end-to-end delay, E2ED). Тогда  $F_S(X)$  будет представлять собой некоторую свёртку представленных показателей. Мы используем мультипликативную функцию [35]. Она базируется на принципе справедливой компенсации отдельных

критериев. Качество маршрута будет оцениваться при помощи метрики (1):

$$F_S(X) = \frac{(1000 * PDR)^{KS1}}{E2ED^{KS2}}, \quad (1)$$

где  $KS1$ ,  $KS2$  весовые коэффициенты соответствующих показателей. Чем больше значение  $PDR$  и чем меньше значение  $E2ED$ , тем качественнее маршрут связи. Для балансировки трафика используем функцию с аналогичной структурой, но с иными весовыми коэффициентами. Изменённые весовые коэффициенты позволят отдать приоритет другому показателю по уже отфильтрованным маршрутам:

$$F_B(X) = \frac{(1000 * PDR)^{KB1}}{E2ED^{KB2}}. \quad (2)$$

Весовые коэффициенты настраиваются исходя из приоритетов критериев эффективности маршрутов, задаваемых пользователями сети. Для более приоритетного критерия выставляется больший вес. Для исключения субъективности в выборе весовых коэффициентов приемлемо использование методик, описанных в [36].

Исходя из формулы (1), маршрут  $R_1$  предпочтительнее  $R_2$ , если  $F_S(X_{R1}) > F_S(X_{R2})$ . При этом маршрут  $R_1$  является пригодным для маршрутизации трафика, если  $F_S(X) > F_S'$ , где  $F_S'$  – некоторое пороговое значение. В нашем случае  $F_S(X)$  – фильтрующая функция. Для балансировки трафика введём функцию  $F_B(X)$  Используется для вычисления пропорций, в каком объёме будет распределён трафик по маршрутам  $R_1, R_2, \dots, R_n$ . Соответственно, трафик будет распределён в пропорциях:

$$F_B(X_{R1}) : F_B(X_{R2}) : \dots : F_B(X_{Rn}). \quad (3)$$

Для оценки  $F_S$  и  $F_B$  мы используем алгоритм градиентного бустинга деревьев принятия решений [37]. Алгоритм представляет способ построения композиции решающих деревьев, где каждое последующее дерево стремится скомпенсировать ошибку уже построенных. Среди преимуществ градиентного бустинга можно выделить отсутствие необходимости нормирования разнородных признаков  $X$ , способность осуществлять работу с произвольными

функциями потерь в задачах регрессии, меньшая подверженность переобучению, достаточная точность восстановления искомым функций.

Рассмотрим процесс обучения градиентного бустинга для функции  $F_S(X)$  с коэффициентами  $KS1 = 0,7$ ,  $KS2 = 0,3$ . В работе [38] показано, что для данной задачи подходят именно данные значения коэффициентов. В ходе работы при помощи имитационной модели беспроводной самоорганизующейся сети был собран набор данных. Параметры моделирования представлены в таблице 1. Каждая запись в наборе представляет массив частных метрик  $X$  производительности маршрута связи, полученных в процессе его построения и является признаковым описанием маршрута. Каждая запись содержит значения  $F_S$ , рассчитанные по результатам функционирования маршрута связи.

Таблица 1. Параметры модели ad-hoc сети при сборе данных

Параметр	Значение
Максимальная длина маршрута	5, 6, 7, ..., 19, 20
Скорость передачи данных	8 Кб/с.
Трафик между абонентами	2 Кб/с.
Максимальная скорость движения абонентов	15 м/с.
Модель движения абонентов	Модель случайных перемещений на основе случайных точек
Радиус радиопередачи	400 м.
Размер карты	1500 x 1500 м <sup>2</sup>
Ограничение по виртуальному времени моделирования	1200 с.

Для оценки возможности построить качественные модели машинного обучения был произведён корреляционный анализ между признаковым описанием маршрута связи и  $F_S$ . Результаты представлены на рисунке 5.  $F_S$  имеет выраженную статистическую взаимосвязь с  $I_{all}$ ,  $L_{av}$ ,  $Q_{av}$ ,  $Delay_{av}$ ,  $R_{ch}$ ,  $R_n$ . Результаты корреляционного анализа демонстрируют возможность построения качественной модели для оценки  $F_S$ .

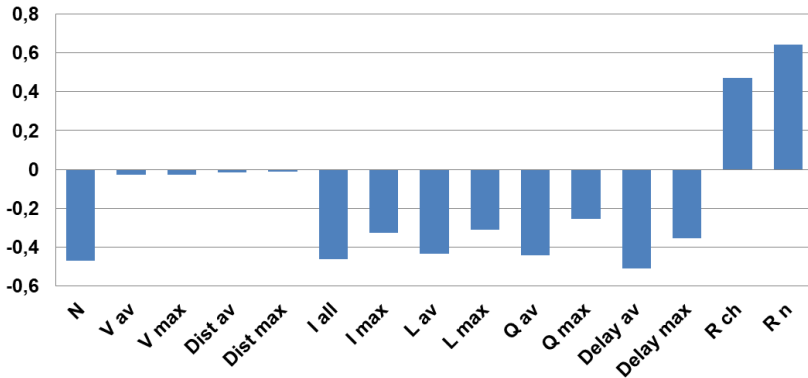


Рис. 5. Корреляционный анализ между признаками  $F_S$

Обучение модели производилось при помощи функционала библиотеки CatBoost [39]. Для проверки эффективности алгоритмов градиентного бустинга набор данных был разделён на обучающуюся ( $X_{train}$ ,  $PDR_{train}$ ,  $E2ED_{train}$ ) и тестовую ( $X_{test}$ ,  $PDR_{test}$ ,  $E2ED_{test}$ ) выборки в пропорции 4:1.

Для оценки эффективности полученной модели использовались метрики  $RMSE$  и  $R^2$ , рассчитанные на тестовой выборке. Были получены значения 4,63 и 0,96 соответственно. Лучшие результаты были получены при количестве деревьев в 100 единиц и глубине деревьев – 5. Близость  $R^2$  к 1,0 и относительно низкое значение  $RMSE$  свидетельствует о высокой эффективности полученной модели.

**3.5. Алгоритм обнаружения маршрутов.** Предлагаемый нами алгоритм построения маршрутов связи между абонентами S и D состоит из двух этапов. На первом этапе (прямой ход) производится широковещательная рассылка пакетов  $RouteSearch$  к узлу D для анализа состояния абонентов и каналов связи. На втором этапе (обратный ход) производится построение маршрутов связи от S к D на основе информации, полученной на первом этапе.

Алгоритм поиска маршрута запускается, когда отсутствует актуальный исправный маршрут к абоненту D. На первом этапе производится накопление актуальной информации о производительности сети, а также формируется децентрализованный граф пересылок между абонентами S и D. В процессе поиска используется пакет  $RouteSearch$  (рисунок 6).



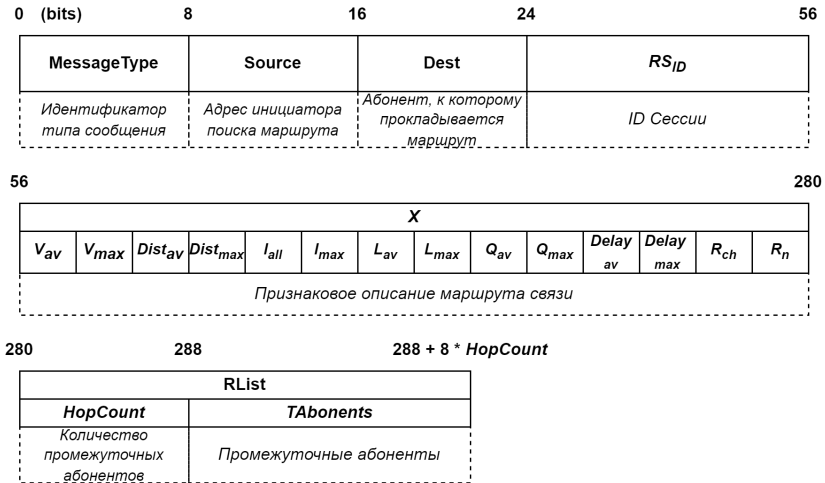


Рис. 6. Структура пакета RouteSearch

Этап прямого хода предназначен для определения возможных направлений маршрутизации трафика между абонентами, а также анализа производительности узлов и каналов связи, которые могут быть включены в маршруты. Функционирование первого этапа производится по алгоритму, представленному на рисунке 7.

Если абоненту S требуется маршрут к абоненту D, он производит широковещательную рассылку пакета RouteSearch. При приёме пакета транзитным абонентом T производится обновление маршрутных метрик X в пакете. Если T обнаруживает в RouteSearch петли, превышение максимально допустимого количества переходов  $HopCount > MaxHopCount$ , совершённых пакетом в сети, пакет уничтожается.

Затем T сохраняет информацию о поступившем пакете в RouteItem абонента D. Если пакет с текущими RS<sub>ID</sub> и S был получен впервые, абонент T производит его ретрансляцию на другие узлы. При этом перед передачей в RouteSearch увеличиваются значение HopCount на 1, а в поле RList добавляется собственный адрес.

Если абонент D принял пакет с текущими RS<sub>ID</sub> и адресом S впервые, то он формирует ответ RouteAnswer с временным маршрутом и возвращает его абоненту S. Временный маршрут используется при передаче данных от S к D до момента, пока не будут сформированы оптимальный и альтернативные маршруты. В течение периода TimeRecvWait абонент D принимает другие пакеты RouteSearch

с аналогичными значениями полей  $RS_{ID}$  и  $S$ . После истечения периода завершается этап прямого хода и запускается процедура обратного хода. Все поступающее после этого пакеты RouteSearch отбрасываются.

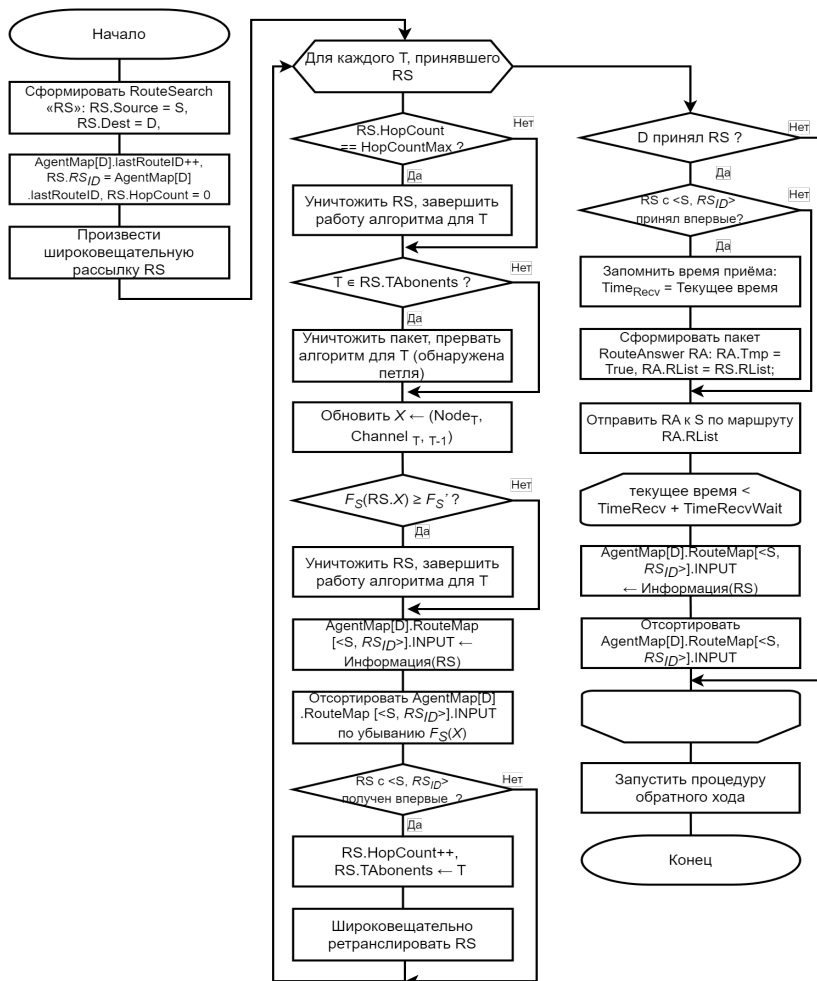


Рис. 7. Алгоритм прямого хода поиска маршрута связи

В результате работы алгоритма каждый транзитный абонент хранит несколько промежуточных маршрутов от  $S$  до себя. Маршруты

отсортированы в порядке убывания  $F_S(X)$ . RouteAnswer с флагом `tmp` содержит временный маршрут связи. Он может быть использован для передачи данных между S и D, пока не будут сформированы постоянные маршруты.

Второй этап инициализируется после завершения процедуры прямого хода. Обратный ход предназначен для формирования оптимального (основного) и альтернативных маршрутов связи. Альтернативные маршруты уступают оптимальному по критерию  $F_S(X)$ , могут частично с ним пересекаться. При этом альтернативные пути приемлемы для маршрутизации трафика (рисунок 8).

В построении маршрутов на этапе обратного хода используется ответный сетевой пакет RouteAnswer (рисунок 9). Он может быть передан в адресном или широковещательном режимах. Адресная передача используется при построении оптимального пути между абонентами S и D. Широковещательная рассылка применяется при определении альтернативных маршрутов.

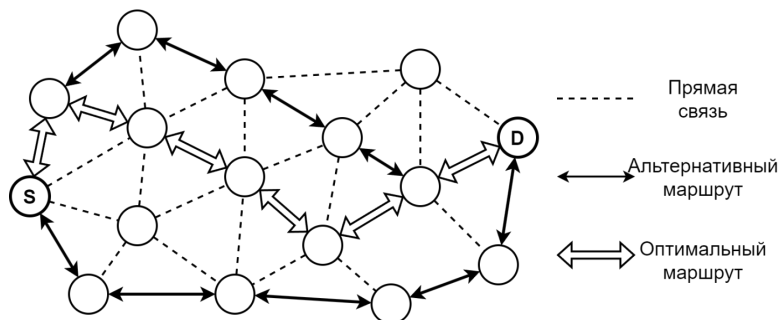


Рис. 8. Пример маршрутов связи, построенных на этапе обратного хода

В процессе обратного хода производится отправка адресного пакета RouteAnswer к абоненту S. В качестве транзитных абонентов выбираются те, с которых поступили лучшие по метрике  $F_S(X)$  частично сформированные маршруты. Транзитные абоненты, через которые прошёл адресный пакет RouteAnswer, добавляются в оптимальный маршрут. Через время `TimeSendWait` после отправки адресного RouteAnswer производится широковещательная рассылка пакета с целью построения альтернативных маршрутов связи. Альтернативный маршрут считается сформированным, если достигает узла S или узла основного маршрута.

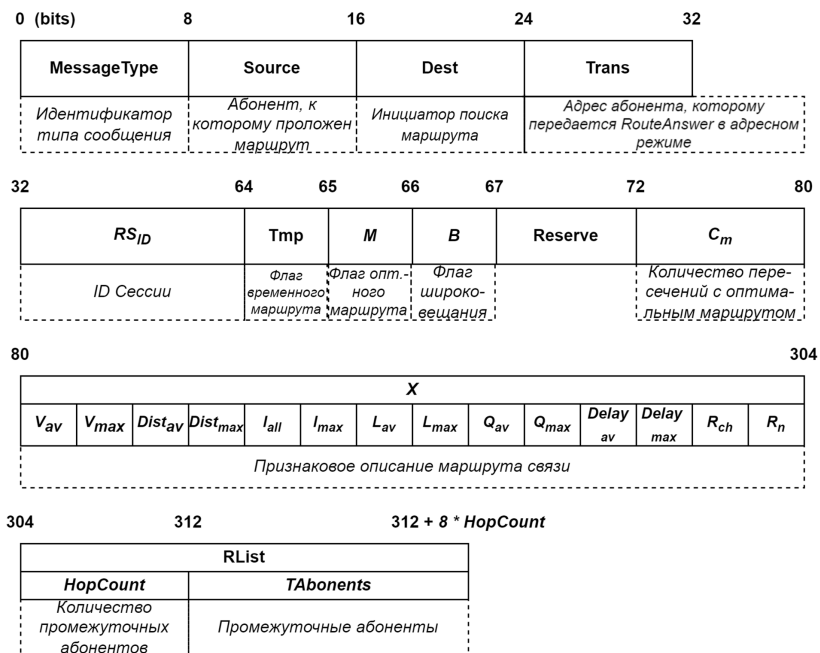


Рис. 9. Структура пакета RouteAnswer

Процедура обратного хода функционирует по алгоритму, представленному на рисунке 10. Также введём параметры *ActiveRouteTime* и *RouteSearchTime*. *ActiveRouteTime* – временной промежуток, в течении которого найденные маршруты считаются актуальными. *RouteSearchTime* – временной период, за который абонент-инициатор поиска маршрута должен получить ответный пакет *RouteAnswer*. В ином случае принимается, что маршрут к целевому абоненту отсутствует.

Если альтернативный маршрут окончательно сформирован на абоненте основного маршрута, то есть альтернативный маршрут имеет пересечение с оптимальным маршрутом, текущий абонент производится отправку пакета *RouteAnswer* по оптимальному маршруту. Маршруты, которые частично пересекаются, будем называть квазипараллельными.

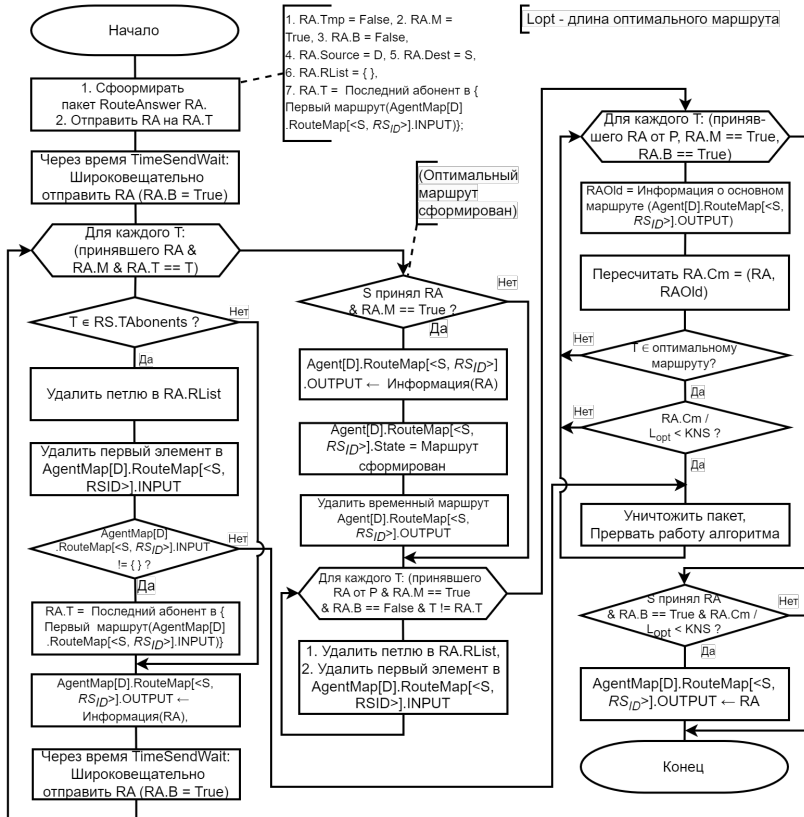


Рис. 10. Алгоритм обратного хода поиска маршрута связи

Результатом функционирования процедур прямого и обратного ходов является построение оптимального и нескольких альтернативных маршрутов между абонентами S и D. Для передачи уведомлений о доставке сообщений используется оптимальный маршрут.

**3.6. Маршрутизация трафика по распределённому маршруту.** Алгоритм доставки данных используется при маршрутизации трафика по найденным маршрутам связи. При передаче сообщений производится их распределение по нескольким актуальным маршрутам в соответствии с критерием  $F_B(X)$ . Критерий  $F_B(X)$  определяет пропорции, в которых трафик будет распределён по найденным маршрутам (рисунок 11). Таким образом, производится балансировка трафика в ad-hoc сети.

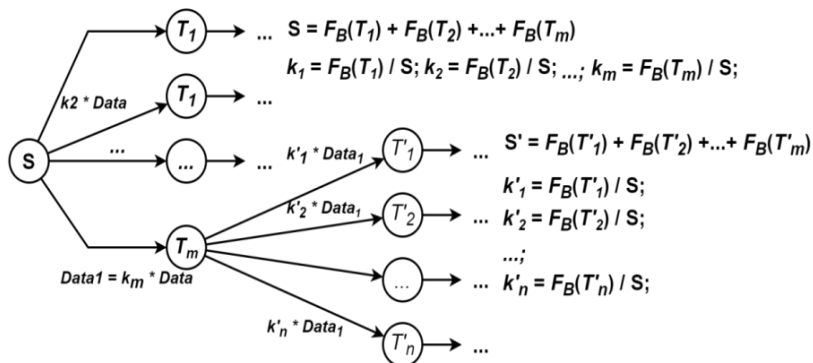


Рис. 11. Схема балансировки трафика

Передача данных производится при помощи сообщений DataRoute, структура которого описана в рисунке 12. Передача данных осуществляется по алгоритму, представленному на рисунке 13. Если целевой абонент D является соседним абонентом 1-го или 2-го порядка для текущего узла, запрос маршрута не требуется. В этом случае, а также при использовании временного маршрута балансировка трафика не производится. Если абонент содержит два и более восходящих маршрута связи, активируется алгоритм балансировки трафика.

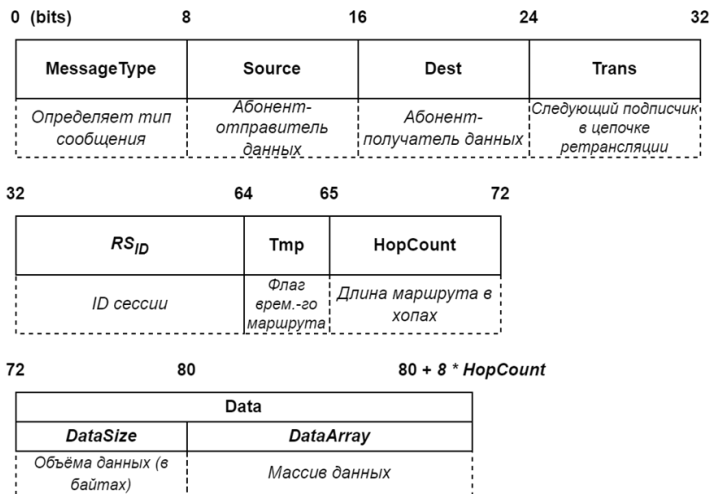


Рис. 12. Структура пакета DataRoute

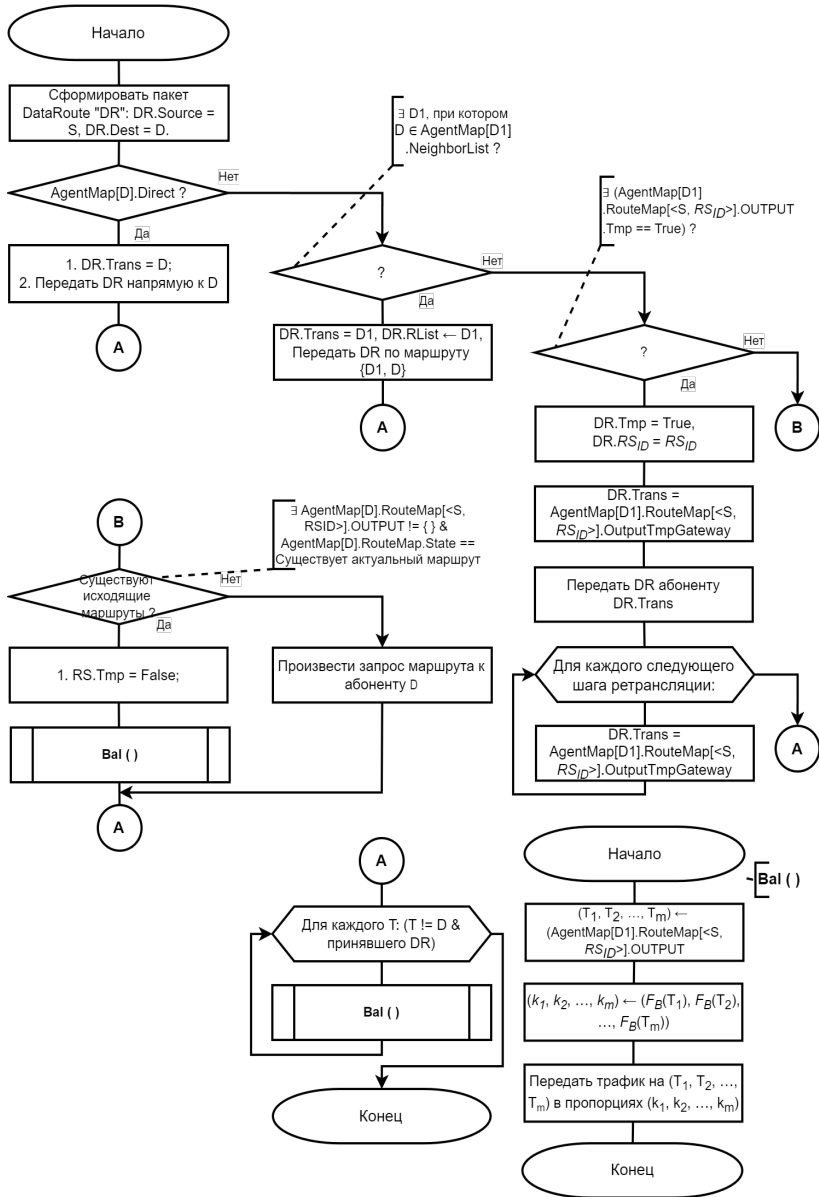


Рис. 13. Алгоритм маршрутизации и балансировки трафика по найденным маршрутам связи

Если по итогу работы поиска маршрутов было получено несколько квазипараллельных маршрутов, трафик распределяется по нескольким путям в соответствии с критерием  $F_B(X)$ . В соответствии с этим производится балансировка трафика по сети. Критерий  $F_B(X)$  определяет пропорции, в которых трафик будет распределяться по найденным маршрутам. При этом распределение трафика по нескольким исходящим каналам осуществляют абоненты, через которые пролегают несколько квазипараллельных маршрутов, и имеется несколько вариантов дальнейшей ретрансляции.

В работе сети возможны случаи, когда между двумя абонентами невозможно построить маршрут. Например, целевой узел D или подсеть, в которой он расположен, не связана с подсетью текущего абонента. Такая ситуация обнаруживается, когда абонент D не является смежным узлом 1-го или 2-го порядка, или когда истекло время RouteSearchTime, и маршрут не был получен. Тогда через время RepeatSearchTime инициируется повторный запрос маршрута связи. Если повторный поиск не принёс результатов, считается, что абонент D не доступен для передачи данных.

**4. Экспериментальное исследование протокола связи.** Для проведения экспериментального исследования была использована имитационная модель ad-hoc сети из [40]. В ходе экспериментального исследования была осуществлена проверка корректности функционирования предложенного в статье протокола связи. Производилось его сравнение с протоколом маршрутизации AOMDV. Несмотря на существование множества его модификаций, они лишь в частных сценариях обеспечивают высокие показатели QoS и в среднем по производительности не превосходят базовый протокол. Другие решения являются коммерческими и требуют специализированного оборудования от конкретных производителей, что снижает возможности по их использованию и исследованию.

В исследовании использовались 3 сценария моделирования, которые отличаются характером мобильности абонентов [41]. В первом сценарии использовалась топология, графически представляющая собой матрицу из абонентов (рисунок 14). У абонентов отсутствует мобильность. Значения надёжности каналов связи имеют случайное значение. Периодически производился разрыв отдельных каналов связи. Статистика рассчитывалась между абонентами S и D. Здесь моделируется взаимодействие между абонентами без мобильности (группы спасателей, стационарные лаборатории).



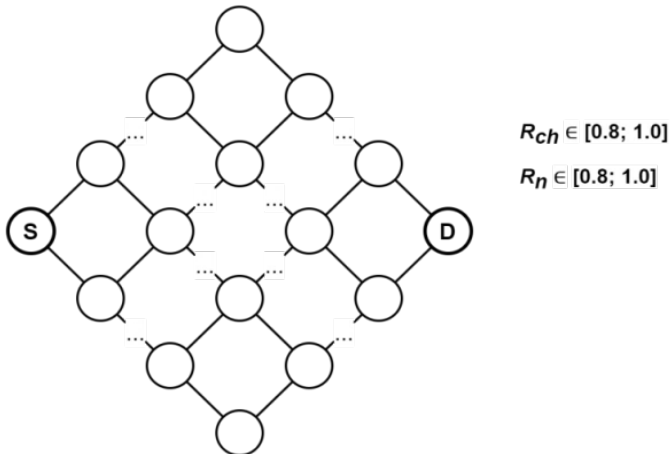


Рис. 14. Топология статичной сети

Следующие эксперименты посвящены проверке алгоритмов на сценариях с высокой динамикой изменения топологии (рисунок 15). Качество и наличие прямых связей между абонентами определяется расстоянием между узлами. Движение абонентов описывается моделью перемещений на основе случайных точек в заданной области. Абоненты отправитель и получатель задаются перед началом моделирования, им присваиваются координаты за пределами зоны пространственного перемещения абонентов. Трафик сети снижен, и передача данных, кроме выделенных узлов, ведётся только между соседними абонентами.

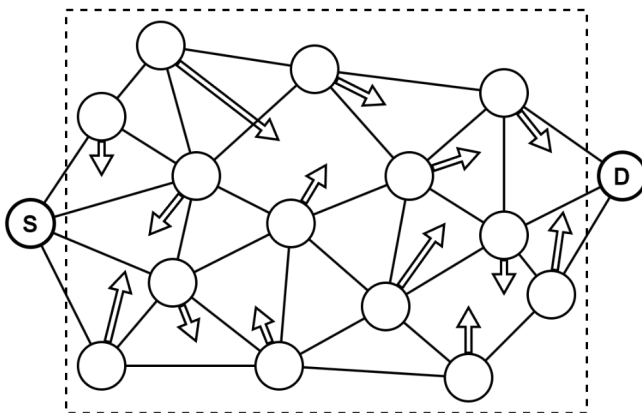


Рис. 15. Топология сети с моделью перемещения абонентов

Во втором эксперименте 80% являются мобильными – скорость их перемещения не превышает 6 м/с. Это могут быть перемещающиеся бригады спасателей, мобильные роботы. В третьем эксперименте 50% мобильных абонентов имеют высокую скорость перемещения. Скорость их перемещения – до 15 м/с. Это может быть спасательная техника в движении или дроны.

В качестве критериев, по которым производилось сравнение протоколов, использовались следующие:

- PDR (packet delivery ratio) – надёжность доставки пакетов данных. Значение критерия показывает долю пакетов, которые достигли целевого абонента;

- E2EDG (group E2ED) – общее время передачи по квазипараллельным маршрутам группы пакетов в составе фрейма.

**4.1. Подготовка к исследованию.** В текущем исследовании скорость передачи данных ровна 64 кбит/с. Прикладной трафик между смежными абонентами составил 8 пакетов секунду, каждый из которых имел объём 255 кб (16 кбит/с). В контрольной передаче данных между двумя абонентами использовались большие фреймы данных по 255 пакетов (фреймы, объёмом 65 025 байт). Другие параметры моделирования представлены в таблице 2. Уровень надёжности каналов связи и среднее время задержки пакетов при их обработке абонентами в одних случаях задавался статически перед началом функционирования сети, в других экспериментах определялся взаимным пространственным расположением абонентов.

В нашей текущей работе в качестве критерия  $F_S(X)$ , предсказываемого при помощи градиентного бустинга, является выражение (1), где  $KS1 = 0,7$ ,  $KS2 = 0,3$ . Аналогичным образом предсказывается метрика  $F_B$ . Всего было подготовлено 4 модели градиентного бустинга (таблица 3). Первая функция представляет собой предсказанное значение надёжности доставки данных по маршруту. Вторая функция ровна используемой  $F_S(X)$ . В третьей критерии PDR и E2ED имеют равный приоритет. В четвёртой приоритет отдаётся критерию E2ED.

Таблица 2. Параметры моделирования

Параметр	Значение
<b>Общие параметры исследования</b>	
Скорость передачи данных	64 Кбит/с.
Трафик между абонентами	16 Кбит/с.
Максимально-допустимая задержка обработки пакетов на абонентах	10 с.
Радиус радиопередачи	400 м.
Ограничение по виртуальному времени моделирования	1200 с.
<b>Параметры первого эксперимента</b>	
Размер карты	1500 x 1500 м <sup>2</sup>
Максимальная длина маршрута	15, 20, 25, 30, 35, 40
Количество абонентов	25, 36, 49, 64, 81, 100
<b>Параметры второго эксперимента</b>	
Модель движения абонентов	Модель перемещений на основе случайных точек
Максимальная скорость движения абонентов с низкой мобильностью	6 м/с.
Размеры карты	1000 x 1000 м <sup>2</sup> , 1200 x 1000 м <sup>2</sup> , 1400 x 1000 м <sup>2</sup> , 1600 x 1000 м <sup>2</sup> , 1800 x 1000 м <sup>2</sup> , 2000 x 1000 м <sup>2</sup>
Максимальная длина маршрута	10, 12, 14, 16, 18, 20
Количество мобильных абонентов	8, 10, 11, 13, 14, 16
<b>Параметры третьего эксперимента</b>	
Модель движения абонентов	Аналогично эксперименту 2
Максимальная скорость движения абонентов с низкой мобильностью	6 м/с.
Максимальная скорость движения абонентов с высокой мобильностью	15 м/с.
Размер карты	Аналогично эксперименту 2
Максимальная длина маршрута	10, 12, 14, 16, 18, 20
Количество мобильных абонентов	8, 10, 11, 13, 14, 16
Количество абонентов с высокой мобильностью	4, 5, 5, 6, 7, 8

Таблица 3. Параметры функции  $F_B(X)$ 

Коэффициент	$F_{B1}(X)$	$F_{B2}(X)$	$F_{B3}(X)$	$F_{B4}(X)$
$KB1$	1	0,7	0,5	0,3
$KB2$	0	0,3	0,5	0,7

**4.2. Результаты экспериментального исследования.** Результаты имитационного моделирования для параметров первого эксперимента представлены на рисунке 16 и рисунке 17. Было получено, что наша реализация ( $F_{B1}$ - $F_{B4}$ ) позволяет существенно повысить надёжности доставки данных по сравнению со стандартным AOMDV. Кроме того, почти в 2 раза удалось сократить время доставки данных за счёт распределения трафика по параллельным маршрутам для всех  $F_B(X)$ .

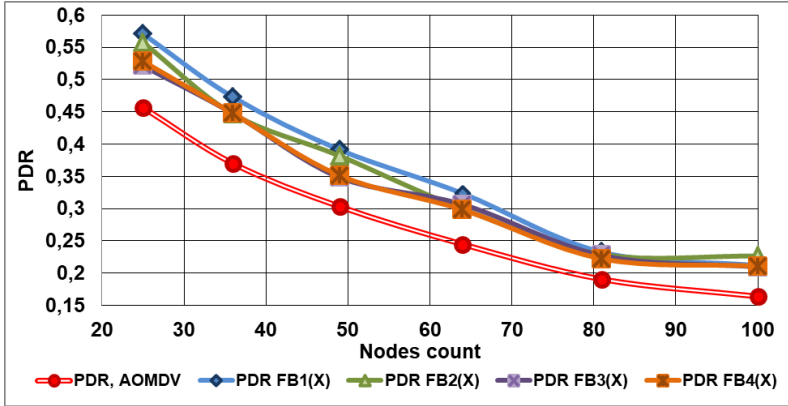


Рис. 16. Результаты экспериментального исследования над метрикой PDR над сценарием со статичной топологией

Результаты экспериментального исследования демонстрируют повышение надёжности доставки данных по сравнению с AOMDV в 1,1–1,4 раз. Повышение показателя PDR обусловлено тем, что в процессе построения маршрутов исключались ненадёжные каналы и перегруженные абоненты сети. За счёт распараллеливания передачи трафика по нескольким маршрутам удалось снизить значение E2EDG – в 2,0–2,2 раз.

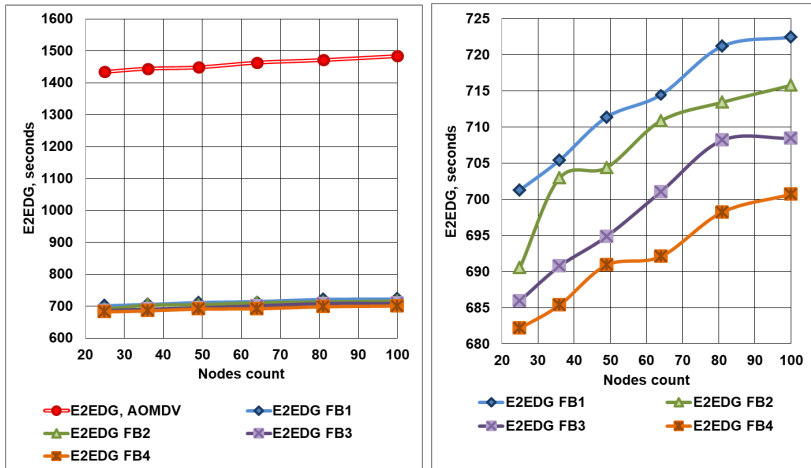


Рис. 17. Результаты экспериментального исследования над метрикой E2EDG над сценарием со статичной топологией

Экспериментальное исследование сети с низким уровнем изменения её динамики представлено на рисунках 18 и 19. Улучшения PDR и E2EDG в целом соответствуют аналогичным показателям в предыдущем эксперименте. PDR улучшен в 1,1-1,2 раза, E2EDG – в 2,0-2,3 раза.

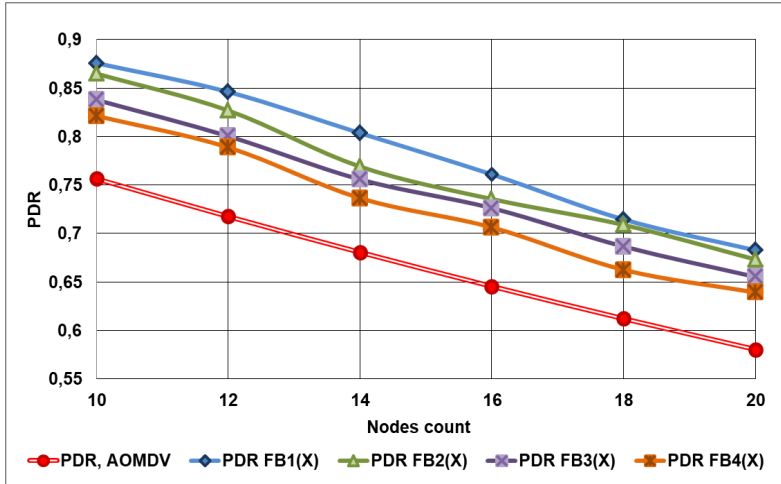


Рис. 18. Результаты экспериментального исследования над метрикой PDR над сценарием с низкодинамичной топологией

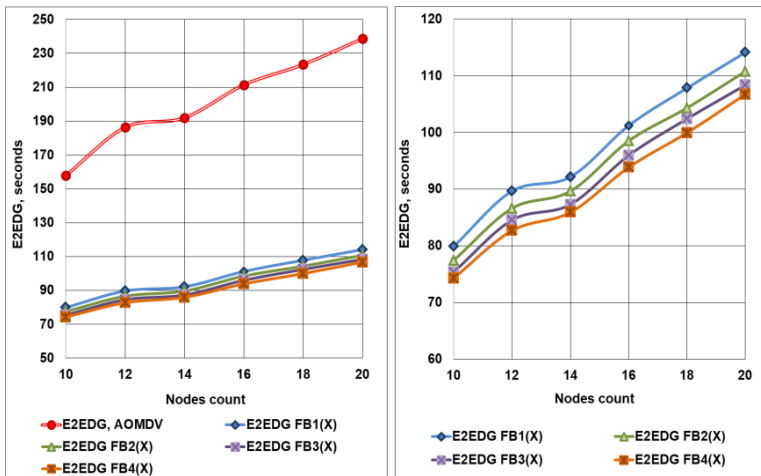


Рис. 19. Результаты экспериментального исследования над метрикой E2EDG над сценарием с низкодинамичной топологией

Результаты эксперимента с высоким уровнем изменения динамики топологии представлено на рисунках 20, 21. Предложенный в работе протокол маршрутизации существенно повышает производительность сети: в 1,2–1,4 для PDR и в 2,1–2,3 для E2EDG.

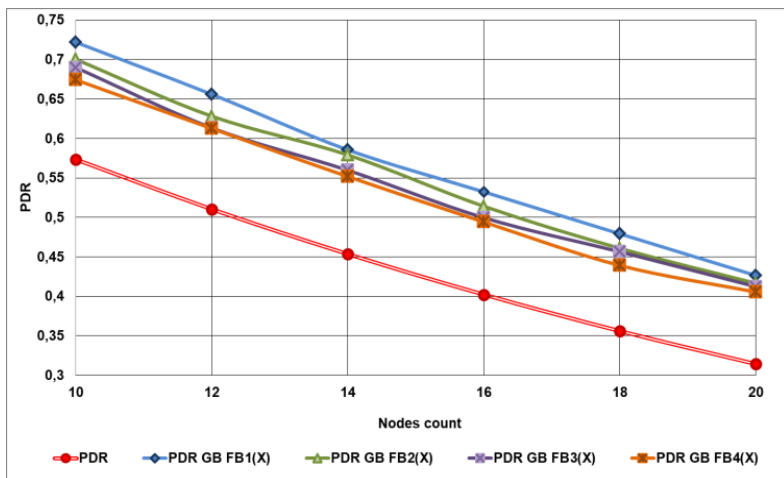


Рис. 20. Результаты экспериментального исследования над метрикой PDR над сценарием с высокодинамичной топологией

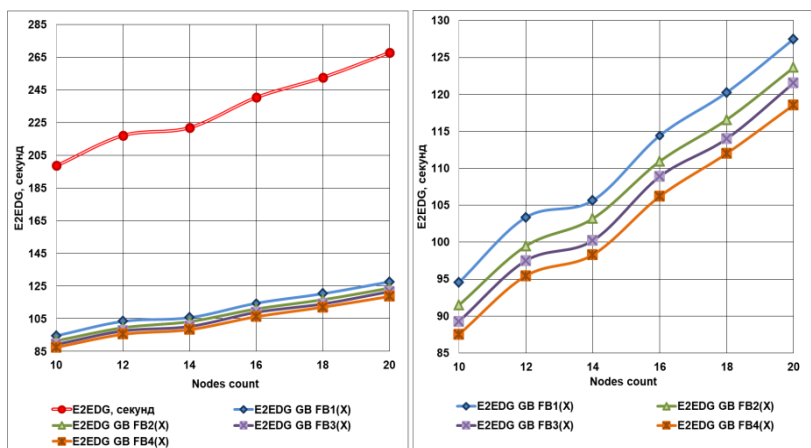


Рис. 21. Результаты экспериментального исследования над метрикой E2EDG над сценарием с высокодинамичной топологией

Результаты экспериментального исследования демонстрируют, что предложенные алгоритмы построения маршрутов и балансировки трафика эффективно работают как на сценариях с низкой динамикой изменения топологии, так и с высокой. При этом во всех случаях одновременно удалось повысить как значение PDR, так и E2EDG. Компромиссным решением, которое позволит в равной степени улучшить оба показателя, является использование функции  $F_{B2}(X)$ .

Таким образом, предложенный протокол позволит ускорить процесс обмена данными между абонентами в сети MANET. Применяемая балансировка трафика позволит разгрузить каналы связи, за счёт чего увеличится пропускная способность сети и повысится QoS. В то же время, это позволит включать в сеть большее количество высокомобильных абонентов с меньшими потерями в QoS и надёжности сети.

Эксперименты подтверждают соответствие протокола требованиям, которые были представлены в начале работы. В частности, повышение надёжности (обеспечения минимального количество потерь пакетов) сопровождалось также и снижением времени доставки данных. Протокол обеспечивает эффективную маршрутизацию трафика для абонентов с различной степенью мобильности. За счёт того, что абоненты не имеют жестко закреплённых за собой функций и балансировка трафика осуществляется децентрализованно, обеспечивается гибкость в построении сети – в неё не требуется вводить специальных узлов маршрутизаторов. Это обеспечит эффективное функционирования сети подразделения аварийной службы.

**5. Заключение.** В работе предложен протокол маршрутизации сетевого уровня, предназначенный для организации децентрализованной связи в подразделениях аварийной службы, где абоненты отличаются разной степенью мобильности и типом передаваемого трафика, который может включать как данные, так и видео- и голосовые сообщения с предъявлением требований к оперативности доставки. Доказано, что для обеспечения высокой надёжности передачи данных в условиях описанных ограничений необходима комплексная оценка маршрута, а также обеспечение возможности одновременной передачи данных по нескольким параллельным маршрутам связи. Были предложены алгоритм обнаружения соседних абонентов, двухэтапный алгоритм построения квазипараллельных маршрутов связи между абонентами и алгоритм балансировки трафика по найденным маршрутам связи. Для оценки маршрутов связи обосновано использование модели градиентного

бустинга. Проведено экспериментальное исследование протокола и выполнено сравнение предложенных алгоритмов с применяющимися в настоящий момент решениями, показавшее, что предлагаемое решение позволит повысить надёжность и оперативность доставки данных, увеличить пропускную способность каналов связи за счет распараллеливания потоков передачи и обеспечить высокое QoS в сети с различной степенью мобильности абонентов. Предлагаемые решения защищены свидетельствами на регистрацию программ для ЭВМ [42, 43, 44].

### Литература

1. Xie Y., Min G., Bin H., Wang H., Ma S. Selection of Communication Frequency Points for Shortwave Radio Network Based on VOACAP // Proceedings of 6th International Conference on Intelligent Computing and Signal (ICSP). 2021. pp. 488–491. DOI: 10.1109/ICSP51882.2021.9408644.
2. Yuxuan G., Yue L., Penghui S. Research Status of Typical Satellite Communication Systems // Proceedings of 19th International Conference on Optical Communications and Networks (ICOON). 2021. pp. 1–3. DOI: 10.1109/ICOON53177.2021.9563909.
3. Duan T., Dinavahi V. Starlink Space Network-Enhanced Cyber-Physical Power System // IEEE Transactions on Smart Grid. 2021. vol. 12. no. 4. pp. 3673–3675. DOI: 10.1109/TSG.2021.3068046.
4. Li S., Zhao Y., He H., Li X., Wang W. The Design and Implementation of Data Link Terminal Based on Two Modes Satellite Communication // Proceedings of International Symposium on Networks, Computers and Communications (ISNCC). 2022. pp. 1–5. DOI: 10.1109/ISNCC55209.2022.9851765.
5. Mohanti S., Bozkaya E., Naderi M.Y., Canberk B., Secinti G., Chowdhury K.R. WiFED Mobile: WiFi Friendly Energy Delivery with Mobile Distributed Beamforming // IEEE/ACM Transactions on Networking. 2021. vol. 29. no. 3. pp. 1362–1375. DOI: 10.1109/TNET.2021.3061082.
6. IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks. Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications // IEEE Std 802.11-2020. 2021. DOI: 10.1109/IEEESTD.2021.9363693.
7. Han J., Geng J., Wu H., Wang K., Zhou H., Ren C., Hayat Q., Zhao X., Yang S., He C., Liang X., Jin R., Ma B., Shen J. The Ultra-Compact ELF Magneto-Mechanical Transmission Antenna with the Speed Modulated EM Signal Based on Three-Phase Induction Motor // IEEE Transactions on Antennas and Propagation. 2021. vol. 69. no. 9. pp. 5286–5296. DOI: 10.1109/TAP.2021.3060149.
8. Hopjan M. Mesh Network Application // Proceeding of International Conference on Military Technologies (ICMT). 2021. pp. 1–4. DOI: 10.1109/ICMT52455.2021.9502829.
9. Wang W., Yang B., Shen S., Shi G., Liu Y. Message Delivery Probability of 3D MANETs Under Erasure Coding // Proceeding International Conference on Networking and Network Applications (NaNA). 2020. pp. 180–185. DOI: 10.1109/NaNA51271.2020.00039.
10. Shruthi S. Proactive routing protocols for a MANET – A review // Proceeding of International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017. pp. 821–827. DOI: 10.1109/I-SMAC.2017.8058294.



11. Ali A.K.S., Kulkarni U.V. Comparing and Analyzing Reactive Routing Protocols (AODV, DSR and TORA) in QoS of MANET // *Proceeding of IEEE 7th International Advance Computing Conference (IACC)*. Hyderabad. India. 2017. pp. 345–348. DOI: 10.1109/IACC.2017.0081.
12. Chai Y., Shi W. Access-enhanced hybrid routing protocol for hybrid wireless mesh network // *IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. Guangzhou. China. 2017. pp. 138–141. DOI: 10.1109/ICCSN.2017.8230094.
13. Ramaiah P., Narmadha R., Gurumoorthy S., Kokulavani K., Gowri V. Exploring Vanet Routing Using A Novel Geographic Routing Protocol // *Proceeding of International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. 2023. pp. 1–6. DOI: 10.1109/ICDCECE57866.2023.10150904.
14. El-Hadidi M.G., Azer M.A. Traffic Analysis for Real Time Applications and its Effect on QoS in MANETs // *Proceeding of International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. 2021. pp. 155–160. DOI: 10.1109/MIUCC52538.2021.9447611.
15. Kiki M.J.M., Iddi I., Yunusa H. Improved AODMV Routing Protocol in Manet UAV Based on Virtual Hop // *Proceeding of 3rd International Conference on Computer Communication and the Internet (ICCCI)*. 2021. pp. 146–151. DOI: 10.1109/ICCCI51764.2021.9486832.
16. Yuan Y.H., Chen H.M., Jia M. An Optimized Ad-hoc On-demand Multipath Distance Vector(AODMV) Routing Protocol // *Proceeding of Asia-Pacific Conference on Communications*. 2005. pp. 569–573. DOI: 10.1109/APCC.2005.1554125.
17. Lee R.H., Jeon D.A. Mobile Ad-hoc Network multi-path routing protocol based on biological attractor selection for disaster recovery communication // *ICT Express*. 2015. vol. 1. no. 2. pp. 86–89. DOI: 10.1016/j.ict.2015.10.001.
18. Tabatabaei S., Nahook H.N. A new routing protocol in MANET using cuckoo optimization algorithm // *Journal of Electrical and Computer Engineering Innovations (JECEI)*. 2020. vol. 9. no. 1. pp. 75–82. DOI: 10.22061/JECEI.2020.7511.397.
19. Subramaniam K., Tamilselvan L. Predictive energy efficient and reliable multicast routing in MANET // *Research Journal of Applied Sciences, Engineering and Technology*. 2015. vol. 9. pp. 706–714. DOI: 10.19026/rjaset.9.2615.
20. Papanna N., Reddy A.R.M., Seetha M. EELAM: Energy efficient lifetime aware multicast route selection for mobile ad hoc networks // *Applied Computing and Informatics*. 2019. vol. 15. pp. 120–128. DOI: 10.1016/j.aci.2017.12.003.
21. Kacem I., Sait B., Mekhilef S., Sabeur N. A New Routing Approach for Mobile Ad Hoc Systems Based on Fuzzy Petri Nets and Ant System // *IEEE Access*. 2018. vol. 6. pp. 65705–65720. DOI: 10.1109/ACCESS.2018.2878145.
22. Bhattacharya A., Sinha K. An efficient protocol for load-balanced multipath routing in mobile ad hoc networks // *Ad Hoc Networks*. 2017. vol. 63. pp. 104–114. DOI: 10.1016/j.adhoc.2017.05.008.
23. Chen J., Li Z., Liu J., Kuo Y. QoS multipath routing protocol based on cross layer design for ad hoc networks // *Proceeding International Conference on Internet Computing and Information Services*. 2011. pp. 261–264. DOI: 10.1109/ICICIS.2011.168.
24. Pandey P., Singh R. Efficient Ad Hoc On Demand Distance Vector Routing Protocol Based on Route Stability in MANETs // *International Journal of Wireless Information Networks*. 2022. vol. 29. no. 3. pp. 393–404. DOI: 10.1007/s10776-022-00570-x.
25. Xiaoxia Q, Wang X., Jiang F. Multi-path Routing Improved Protocol in AODV Based on Nodes Energy // *International Journal of Future Generation Communication and Networking*. 2015. vol. 8. no. 1. pp. 207–214. DOI: 10.14257/ijfgcn.2015.8.1.21.

26. Alghamdi S.A. Load balancing maximal minimal nodal residual energy ad hoc on-demand multipath distance vector routing protocol (LBMMRE-AOMDV) // *Wireless Netw.* 2016. vol. 22. pp. 1355–1363. DOI: 10.1007/s11276-015-1029-6.
27. Er-rouidi M., Moudni H., Mouncif H., Merbouha A. A balanced energy consumption in mobile ad hoc network // *Procedia Computer Science.* 2019. vol. 151. pp. 1182–1187. DOI: 10.1016/j.procs.2019.04.169.
28. Anand M., Sasikala T. Efficient energy optimization in mobile ad hoc network (MANET) using better-quality AODV protocol // *Cluster Computing.* 2019. vol. 22. pp. 12681–12687. DOI: 10.1007/s10586-018-1721-2.
29. Periyasamy P., Karthikeyan E. Link reliable multipath routing protocol for mobile ad hoc networks // *Proceeding of International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015].* 2015. pp. 1–7. DOI: 10.1109/ICCPCT.2015.7159291.
30. Benatia S.E., Smail O., Boudjelal M., Cousin B. ESMRsc: Energy Aware and Stable Multipath Routing Protocol for Ad Hoc Networks in Smart City // *Proceeding of Renewable Energy for Smart and Sustainable Cities.* 2019. pp. 31–42.
31. Rump F., Jopen S.A., Frank M. Using Probabilistic Multipath Routing to Improve Route Stability in MANETs // *Proceeding IEEE 41st Conference on Local Computer Networks (LCN), Dubai, United Arab Emirates.* 2016. pp. 192–195. DOI: 10.1109/LCN.2016.40.
32. Gomes R.D., Queiroz D.V., Lima A.C., Fonseca I.E., Alencar M.S. Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes // *Ad Hoc Networks.* 2017. vol. 59. pp. 116–133. DOI: 10.1016/j.adhoc.2017.02.007.
33. Alghamdi S.A. Load balancing ad hoc on-demand multipath distance vector (LBAOMDV) routing protocol // *EURASIP Journal on Wireless Communications and Networking.* 2015. no. 242. DOI: 10.1186/s13638-015-0453-8.
34. Pourbemany J., Mirjalily G., Abouei J., Fahim Raouf A.H. Load Balanced Ad-Hoc On-Demand Routing Based on Weighted Mean Queue Length Metric // *Proceeding of Electrical Engineering (ICEE).* 2018. pp. 470–475. DOI: 10.1109/ICEE.2018.8472705.
35. Novikov A.S., Ivutin A., Voloshko A., Pestin M.S. Method for Optimizing Ad-hoc Networks Communication Protocol Parameter Values // *Proceeding of 9th Mediterranean Conference on Embedded Computing (MECO).* 2020. pp. 1–4. DOI: 10.1109/MECO49872.2020.9134154.
36. Постников В.М., Спиридонов С.Б. Методы выбора весовых коэффициентов локальных критериев // *Наука и Образование.* 2015. № 06. С. 267–287. DOI: 10.7463/0615.0780334.
37. Piryonesi S.M., El-Diraby T. Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index // *Journal of Infrastructure Systems.* 2020. vol. 26. no. 1. 25 p. DOI: 10.1061/(ASCE)IS.1943-555X.0000512.
38. Pestin M.S., Novikov A.S. Protocol for Multipath Routing of Traffic in Wireless Ad-Hoc Networks Based on the Status of Channels and Network Nodes // *International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation.* 2022. pp. 553–558. DOI: 10.1109/RusAutoCon54946.2022.9896315.
39. Jhaveri S., Khedkar I., Kantharia Y., Jaswal S. Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns // *Proceeding of 3rd International Conference on Computing Methodologies and Communication (ICCMC).* 2019. pp. 1170–1173. DOI: 10.1109/ICCMC.2019.8819828.
40. Пестин М.С., Новиков А.С. Имитационная модель беспроводной ad-hoc сети для исследования алгоритмов маршрутизации трафика // *Прикладная информатика.* 2022. Т. 17. № 4. С. 75–86. DOI: 10.37791/2687-0649-2022-17-4-75-86.
41. Pullin A., Pattinson C., Kor A.L. Building Realistic Mobility Models for Mobile Ad Hoc Networks // *Informatics.* 2018. vol. 5. no. 2. p. 22. DOI: 10.3390/informatics5020022.

42. Пестин М.С., Новиков А.С. Программа для исследования характеристик протоколов маршрутизации в беспроводных децентрализованных сетях связи // Свидетельство о регистрации программы для ЭВМ RU 2022612832. 2022.
43. Пестин М.С., Новиков А.С. Программное обеспечение для моделирования взаимного пространственного перемещения отдельных узлов в беспроводных децентрализованных сетях связи // Свидетельство о регистрации программы для ЭВМ RU 2022663501. 2022.
44. Пестин М.С., Новиков А.С. Программное обеспечение для маршрутизации трафика в беспроводных децентрализованных сетях связи // Свидетельство о регистрации программы для ЭВМ RU 2022663502. 2022.

**Ивутин Алексей Николаевич** — д-р техн. наук, профессор, заведующий кафедрой, кафедра вычислительной техники, Тульский государственный университет (ТулГУ). Область научных интересов: многоагентные системы, учебно-тренировочные средства, параллельные и распределенные информационные процессы. Число научных публикаций — 160. alexey.ivutin@gmail.com; проспект Ленина, 92, 300012, Тула, Россия; р.т.: +7(4872)257-943.

**Новиков Александр Сергеевич** — канд. техн. наук, доцент кафедры, кафедра вычислительной техники, Тульский государственный университет (ТулГУ). Область научных интересов: сетевые протоколы, обеспечение устойчивой связи в беспроводных децентрализованных сетях, статический и динамический анализ программного кода, поиск гонок данных в программном обеспечении, распараллеливание программного кода. Число научных публикаций — 87. thesis-tsu@yandex.ru; проспект Ленина, 92, 300012, Тула, Россия; р.т.: +7(4872)257-943.

**Пестин Максим Сергеевич** — аспирант, кафедра вычислительной техники, Тульский государственный университет. Область научных интересов: взаимодействие программ и программных систем, беспроводные децентрализованные самоорганизующиеся сети. Число научных публикаций — 25. maxime1996rus@mail.ru; проспект Ленина, 92, 300012, Тула, Россия; р.т.: +7(4872)257-943.

**Волошко Анна Геннадьевна** — канд. техн. наук, доцент кафедры, кафедра вычислительной техники, Тульский государственный университет (ТулГУ). Область научных интересов: сетевые протоколы, моделирование параллельных систем, статический и динамический анализ программного кода, поиск гонок данных в программном обеспечении, распараллеливание программного кода. Число научных публикаций — 91. atroshina@mail.ru; проспект Ленина, 92, 300012, Тула, Россия; р.т.: +7(4872)257-943.

A. IVUTIN, A. NOVIKOV, M. PESTIN, A. VOLOSHKO  
**DECENTRALIZED PROTOCOL FOR ORGANIZING  
SUSTAINABLE INTERACTION BETWEEN SUBSCRIBERS  
IN NETWORKS WITH HIGH DYNAMICS OF TOPOLOGY  
CHANGES**

*Ivutin A., Novikov A., Pestin M., Voloshko A.* **Decentralized Protocol for Organizing Sustainable Interaction between Subscribers in Networks with High Dynamics of Topology Changes.**

**Abstract.** Emergency services often have to carry out rescue and liquidation operations in the absence of centralized communications. The inability to ensure stable communication between members of the rescue unit significantly reduces the quality of work. Moreover, in modern realities, stable communication means not only voice exchange, which can be provided by shortwave radio transmitters, but also intensive exchange of large volumes of traffic. The use of standard solutions based on standard network equipment (Wi-Fi, satellite communications, etc.) and existing algorithms for ensuring quality of service in the conditions under consideration does not allow quickly ensuring information exchange between heterogeneous subscribers. Moreover, operation in high-Hz bands can be very difficult in the presence of obstacles, which reduces the overall coverage area and the quality of data transmission. We propose a network layer routing protocol designed to organize decentralized communication in an emergency service department, where subscribers have different degrees of mobility and types of transmitted traffic. This protocol includes algorithms for connecting to the network, detecting optimal and alternative communication routes, and transmitting and balancing traffic along the found routes. The original route search algorithm analyzes the performance of communication channels and determines all possible paths for transmitting traffic between subscribers. Using the route evaluation function based on gradient boosting of decision trees, optimal and alternative communication routes are formed, and when transmitting data, traffic balancing is performed based on the received information. An experimental study of the proposed protocol showed an improvement in the speed of deployment and quality of service in scenarios with varying degrees of subscriber mobility.

**Keywords:** MANET, ad-hoc routing protocol, routing metrics, gradient boosting, packet delivery ratio, end-to-end delay, quality of service.

## References

1. Xie Y., Min G., Bin H., Wang H., Ma S. Selection of Communication Frequency Points for Shortwave Radio Network Based on VOACAP. Proceedings of 6th International Conference on Intelligent Computing and Signal (ICSP). 2021. pp. 488–491. DOI: 10.1109/ICSP51882.2021.9408644.
2. Yuxuan G., Yue L., Penghui S. Research Status of Typical Satellite Communication Systems. Proceedings of 19th International Conference on Optical Communications and Networks (ICOON). 2021. pp. 1–3. DOI: 10.1109/ICOON53177.2021.9563909.
3. Duan T., Dinavahi V. Starlink Space Network-Enhanced Cyber-Physical Power System. IEEE Transactions on Smart Grid. 2021. vol. 12. no. 4. pp. 3673–3675. DOI: 10.1109/TSG.2021.3068046.
4. Li S., Zhao Y., He H., Li X., Wang W. The Design and Implementation of Data Link Terminal Based on Two Modes Satellite Communication. Proceedings of International Symposium on Networks, Computers and Communications (ISNCC). 2022. pp. 1–5. DOI: 10.1109/ISNCC55209.2022.9851765.

5. Mohanti S., Bozkaya E., Naderi M.Y., Canberk B., Secinti G., Chowdhury K.R. WiFED Mobile: WiFi Friendly Energy Delivery with Mobile Distributed Beamforming. *IEEE/ACM Transactions on Networking*. 2021. vol. 29. no. 3. pp. 1362–1375. DOI: 10.1109/TNET.2021.3061082.
6. IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks. Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2020*. 2021. DOI: 10.1109/IEEESTD.2021.9363693.
7. Han J., Geng J., Wu H., Wang K., Zhou H., Ren C., Hayat Q., Zhao X., Yang S., He C., Liang X., Jin R., Ma B., Shen J. The Ultra-Compact ELF Magneto-Mechanical Transmission Antenna with the Speed Modulated EM Signal Based on Three-Phase Induction Motor. *IEEE Transactions on Antennas and Propagation*. 2021. vol. 69. no. 9. pp. 5286–5296. DOI: 10.1109/TAP.2021.3060149.
8. Hopjan M. Mesh Network Application. *Proceeding of International Conference on Military Technologies (ICMT)*. 2021. pp. 1–4. DOI: 10.1109/ICMT52455.2021.9502829.
9. Wang W., Yang B., Shen S., Shi G., Liu Y. Message Delivery Probability of 3D MANETs Under Erasure Coding. *Proceeding International Conference on Networking and Network Applications (NaNA)*. 2020. pp. 180–185. DOI: 10.1109/NaNA51271.2020.00039.
10. Shruthi S. Proactive routing protocols for a MANET – A review. *Proceeding of International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2017. pp. 821–827. DOI: 10.1109/I-SMAC.2017.8058294.
11. Ali A.K.S., Kulkarni U.V. Comparing and Analyzing Reactive Routing Protocols (AODV, DSR and TORA) in QoS of MANET. *Proceeding of IEEE 7th International Advance Computing Conference (IACC)*. Hyderabad. India. 2017. pp. 345–348. DOI: 10.1109/IACC.2017.0081.
12. Chai Y., Shi W. Access-enhanced hybrid routing protocol for hybrid wireless mesh network. *IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. Guangzhou. China. 2017. pp. 138–141. DOI: 10.1109/ICCSN.2017.8230094.
13. Ramaiah P., Narmadha R., Gurumoorthy S., Kokulavani K., Gowri V. Exploring Vanet Routing Using A Novel Geographic Routing Protocol. *Proceeding of International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. 2023. pp. 1–6. DOI: 10.1109/ICDCECE57866.2023.10150904.
14. El-Hadidi M.G., Azer M.A. Traffic Analysis for Real Time Applications and its Effect on QoS in MANETs. *Proceeding of International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. 2021. pp. 155–160. DOI: 10.1109/MIUCC52538.2021.9447611.
15. Kiki M.J.M., Iddi I., Yunusa H. Improved AOMDV Routing Protocol in Manet UAV Based on Virtual Hop. *Proceeding of 3rd International Conference on Computer Communication and the Internet (ICCCI)*. 2021. pp. 146–151. DOI: 10.1109/ICCCI51764.2021.9486832.
16. Yuan Y.H., Chen H.M., Jia M. An Optimized Ad-hoc On-demand Multipath Distance Vector(AOMDV) Routing Protocol. *Proceeding of Asia-Pacific Conference on Communications*. 2005. pp. 569–573. DOI: 10.1109/APCC.2005.1554125.
17. Lee R.H., Jeon D.A. Mobile Ad-hoc Network multi-path routing protocol based on biological attractor selection for disaster recovery communication. *ICT Express*. 2015. vol. 1. no. 2. pp. 86–89. DOI: 10.1016/j.ict.2015.10.001.

18. Tabatabaei S., Nahook H.N. A new routing protocol in MANET using cuckoo optimization algorithm. *Journal of Electrical and Computer Engineering Innovations (JECEI)*. 2020. vol. 9. no. 1. pp. 75–82. DOI: 10.22061/JECEI.2020.7511.397.
19. Subramaniam K., Tamilselvan L. Predictive energy efficient and reliable multicast routing in MANET. *Research Journal of Applied Sciences, Engineering and Technology*. 2015. vol. 9. pp. 706–714. DOI: 10.19026/rjaset.9.2615.
20. Papanna N., Reddy A.R.M., Seetha M. EELAM: Energy efficient lifetime aware multicast route selection for mobile ad hoc networks. *Applied Computing and Informatics*. 2019. vol. 15. pp. 120–128. DOI: 10.1016/j.aci.2017.12.003.
21. Kacem I., Sait B., Mekhilef S., Sabeur N. A New Routing Approach for Mobile Ad Hoc Systems Based on Fuzzy Petri Nets and Ant System. *IEEE Access*. 2018. vol. 6. pp. 65705–65720. DOI: 10.1109/ACCESS.2018.2878145.
22. Bhattacharya A., Sinha K. An efficient protocol for load-balanced multipath routing in mobile ad hoc networks. *Ad Hoc Networks*. 2017. vol. 63. pp. 104–114. DOI: 10.1016/j.adhoc.2017.05.008.
23. Chen J., Li Z., Liu J., Kuo Y. QoS multipath routing protocol based on cross layer design for ad hoc networks. *Proceeding International Conference on Internet Computing and Information Services*. 2011. pp. 261–264. DOI: 10.1109/ICICIS.2011.168.
24. Pandey P., Singh R. Efficient Ad Hoc On Demand Distance Vector Routing Protocol Based on Route Stability in MANETs. *International Journal of Wireless Information Networks*. 2022. vol. 29. no. 3. pp. 393–404. DOI: 10.1007/s10776-022-00570-x.
25. Xiaoxia Q, Wang X., Jiang F. Multi-path Routing Improved Protocol in AODV Based on Nodes Energy. *International Journal of Future Generation Communication and Networking*. 2015. vol. 8. no. 1. pp. 207–214. DOI: 10.14257/ijfgen.2015.8.1.21.
26. Alghamdi S.A. Load balancing maximal minimal nodal residual energy ad hoc on-demand multipath distance vector routing protocol (LBMMRE-AOMDV). *Wireless Netw.* 2016. vol. 22. pp. 1355–1363. DOI: 10.1007/s11276-015-1029-6.
27. Er-rouidi M., Moudni H., Mouncif H., Merbouha A. A balanced energy consumption in mobile ad hoc network. *Procedia Computer Science*. 2019. vol. 151. pp. 1182–1187. DOI: 10.1016/j.procs.2019.04.169.
28. Anand M., Sasikala T. Efficient energy optimization in mobile ad hoc network (MANET) using better-quality AODV protocol. *Cluster Computing*. 2019. vol. 22. pp. 12681–12687. DOI: 10.1007/s10586-018-1721-2.
29. Periyasamy P., Karthikeyan E. Link reliable multipath routing protocol for mobile ad hoc networks. *Proceeding of International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. 2015. pp. 1–7. DOI: 10.1109/ICCPCT.2015.7159291.
30. Benatia S.E., Smail O., Boudjelal M., Cousin B. ESMRsc: Energy Aware and Stable Multipath Routing Protocol for Ad Hoc Networks in Smart City. *Proceeding of Renewable Energy for Smart and Sustainable Cities*. 2019. pp. 31–42.
31. Rump F., Jopen S.A., Frank M. Using Probabilistic Multipath Routing to Improve Route Stability in MANETs. *Proceeding IEEE 41st Conference on Local Computer Networks (LCN)*, Dubai, United Arab Emirates. 2016. pp. 192–195. DOI: 10.1109/LCN.2016.40.
32. Gomes R.D., Queiroz D.V., Lima A.C., Fonseca I.E., Alencar M.S. Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes. *Ad Hoc Networks*. 2017. vol. 59. pp. 116–133. DOI: 10.1016/j.adhoc.2017.02.007.
33. Alghamdi S.A. Load balancing ad hoc on-demand multipath distance vector (LBAOMDV) routing protocol // *EURASIP Journal on Wireless Communications and Networking*. 2015. no. 242. DOI: 10.1186/s13638-015-0453-8.
34. Pourbemany J., Mirjalily G., Abouei J., Fahim Raouf A.H. Load Balanced Ad-Hoc On-Demand Routing Based on Weighted Mean Queue Length Metric. *Proceeding of Electrical Engineering (ICEE)*. 2018. pp. 470–475. DOI: 10.1109/ICEE.2018.8472705.

35. Novikov A.S., Ivutin A., Voloshko A., Pestin M.S. Method for Optimizing Ad-hoc Networks Communication Protocol Parameter Values. Proceeding of 9th Mediterranean Conference on Embedded Computing (MECO). 2020. pp. 1–4. DOI: 10.1109/MECO49872.2020.9134154.
36. Postnikov V.M., Spiridonov S.B. [Selecting methods of the weighting factors of local criteria]. Nauka i Obrazovanie – Science and Education. 2015. no. 06. pp. 267–287. DOI: 10.7463/0615.0780334. (In Russ.).
37. Piryonesi S.M., El-Diraby T. Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index. Journal of Infrastructure Systems. 2020. vol. 26. no. 1. 25 p. DOI: 10.1061/(ASCE)IS.1943-555X.0000512.
38. Pestin M.S., Novikov A.S. Protocol for Multipath Routing of Traffic in Wireless Ad-Hoc Networks Based on the Status of Channels and Network Nodes. International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation. 2022. pp. 553–558. DOI: 10.1109/RusAutoCon54946.2022.9896315.
39. Jhaveri S., Khedkar I., Kantharia Y., Jaswal S. Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns. Proceeding of 3rd International Conference on Computing Methodologies and Communication (ICCMC). 2019. pp. 1170–1173. DOI: 10.1109/ICCMC.2019.8819828.
40. Pestin M., Novikov A. [Simulation model of wireless ad-hoc network to study algorithms of traffic routing]. Prikladnaya informatika – Journal of Applied Informatics. 2022. vol. 17. no. 4. pp. 75–86. DOI: 10.37791/2687-0649-2022-17-4-75-86. (In Russ.).
41. Pullin A., Pattinson C., Kor A.L. Building Realistic Mobility Models for Mobile Ad Hoc Networks. Informatics. 2018. vol. 5. no. 2. p. 22. DOI: 10.3390/informatics5020022.
42. Pestin M.S., Novikov A.S. Programma dlia issledovaniia kharakteristik protokolov marshrutizatsii v besprovodnykh detsentralizovannykh setiakh sviazi [Software for studying the characteristics of routing protocols in wireless decentralized communication networks]. Certificate of registration of a computer program, no. 2022612832. 2022. (In Russ.).
43. Pestin M.S., Novikov A.S. Programmnoe obespechenie dlia modelirovaniia vzaimnogo prostranstvennogo peremeshcheniia otdel'nykh uzlov v besprovodnykh detsentralizovannykh setiakh sviazi [Software for modeling the mutual spatial movement of individual nodes in wireless decentralized communication networks]. Certificate of registration of a computer program, no. 2022663501. 2022. (In Russ.).
44. Pestin M. S., Novikov A. S. Programmnoe obespechenie dlia marshrutizatsii-trafika v besprovodnykh detsentralizovannykh setiakh sviazi [Software for Traffic Routing in Wireless Decentralized Communication Networks]. Certificate of registration of a computer program, no. 2022663502. 2022. (In Russ.).

**Ivutin Alexey** — Ph.D., Dr.Sci., Professor, Head of the department, Department of computer technology, Tula State University. Research interests: multi-agent systems, training aids and simulators, parallel and distributed information processes. The number of publications — 160. alexey.ivutin@gmail.com; 92, Lenin Av., 300012, Tula, Russia; office phone: +7(4872)257-943.

**Novikov Alexander** — Ph.D., Associate professor of the department, Department of computer technology, Tula State University. Research interests: network protocols, ensuring stable communications in wireless decentralized networks, static and dynamic analysis of program code, data races detection in software, parallelizing program code. The number of publications — 87. thesis-tsu@yandex.ru; 92, Lenin Av., 300012, Tula, Russia; office phone: +7(4872)257-943.

**Pestín Maxim** — Postgraduate, Department of computer technology, Tula state university. Research interests: interaction of programs and software systems, wireless decentralized self-organizing networks. The number of publications — 25. maxime1996rus@mail.ru; 92, Lenin Av., 300012, Tula, Russia; office phone: +7(4872)257-943.

**Voloshko Anna** — Ph.D., Associate professor of the department, Department of computer technology, Tula State University. Research interests: network protocols, simulation of parallel systems, static and dynamic analysis of program code, data race detection in software, parallelizing program code. The number of publications — 91. atroshina@mail.ru; 92, Lenin Av., 300012, Tula, Russia; office phone: +7(4872)257-943.



В.Н. ЯКИМОВ  
**ВОССТАНОВЛЕНИЕ ДИСКРЕТНОЙ  
ПОСЛЕДОВАТЕЛЬНОСТИ СИГНАЛА НА ОСНОВЕ МОДЕЛИ  
СКОЛЬЗЯЩЕГО СРЕДНЕГО И ОЦЕНКИ КОРРЕЛЯЦИОННОЙ  
СВЯЗИ ОТСЧЕТОВ ПРИ ПРЯМОМ И ОБРАТНОМ  
ПРОГНОЗИРОВАНИИ**

*Якимов В.Н.* Восстановление дискретной последовательности сигнала на основе модели скользящего среднего и оценки корреляционной связи отсчетов при прямом и обратном прогнозировании.

**Аннотация.** В статье рассмотрена разработка математического обеспечения для восстановления значений отсчетов дискретной последовательности, которая была получена в результате равномерной дискретизации непрерывного во времени сигнала. Задача восстановления решается исходя из того, что сигнал можно рассматривать как стационарный или стационарный хотя бы в широком смысле (квазистационарный). Разработка математического обеспечения для восстановления значений отсчетов сигнала осуществлена на основе построения модели скользящего среднего и оценки корреляционной связи отсчетов сигнала во времени при прямом и обратном прогнозировании. Необходимая для восстановления значений отсчетов выборка оценок корреляционной функции сигнала вычисляется по отсчетам с известными значениями. С учетом выполнения условия стационарности сигнала это можно сделать на любом участке последовательности независимо от места нахождения восстанавливаемого участка. Полученные оценки отсчетов корреляционной функции могут использоваться как для прямого, так и для обратного прогнозирования. При этом даже если необходимо восстановить несколько проблемных участков, достаточно только один раз вычислить необходимую для их восстановления выборку оценок корреляционной функции. На основе полученного математического решения поставленной задачи разработано алгоритмическое обеспечение. Тестовые испытания и функциональные проверки алгоритмического обеспечения были осуществлены на основе имитационного моделирования с использованием модели сигнала, представляющей собой аддитивную сумму гармонических компонент со случайными начальными фазами. Полученные результаты показали, что вычисление оценок значений утраченных отсчетов осуществляется с достаточно низкой погрешностью, как при прямом, так и при обратном прогнозировании, а также при их совместном использовании. На практике выбор алгоритма восстановления последовательности на основе прямого или обратного прогнозирования будет определяться исходя из реальных условий. В частности, если предыдущих отсчетов с известными значениями недостаточно для прямого прогнозирования, то осуществляется процедура обратного прогнозирования и наоборот. Разработанное алгоритмическое обеспечение может быть реализовано в виде метрологически значимого программного обеспечения для многофункциональных систем цифровой обработки сигналов.

**Ключевые слова:** сигналы дискретного времени, последовательность отсчетов, восстановление сигнала, прямое и обратное прогнозирование, корреляционная связь.

**1. Введение.** Восстановление поврежденных участков дискретной последовательности сигнала по конечному числу отсчетов с известными значениями является актуальной задачей во многих

научных и прикладных областях исследований. В частности, особое внимание восстановлению утраченных значений отсчетов уделяется при рассмотрении практических ситуаций, возникающих при цифровой обработке сигналов в радиолокации, вибродиагностике, беспроводной передаче данных и т.д. [1 – 10].

Дискретные последовательности сигналов приходится восстанавливать из-за сбоев в работе их источников или средств аналого-цифрового преобразования принимающей стороны [11 – 13]. Искажение значений отсчетов на отдельных участках последовательностей может быть вызвано импульсными шумами (скачкообразным изменением фоновых шумов) [14]. В отдельных случаях дискретная последовательность может иметь участки с непостоянной временной сеткой, например, из-за возникновения краткосрочного эффекта дрожания периода тактовых импульсов синхронизирующих устройств [15 – 18]. Может иметь место разреженность отсчетов [19]. Дискретная последовательность может быть преднамеренно сформирована с нерегулярным интервалом дискретизации или в результате снижения числа уровней квантования [20 – 25]. Неравномерность временной сетки дискретной последовательности может наблюдаться в высокоскоростных многоканальных системах обработки сигналов с использованием параллельно работающих низкоскоростных аналого-цифровых преобразователей, когда объединение параллельно формируемых выборок может осуществляться со смещением и потерей отсчетов [26 – 28].

При разработки алгоритмов восстановления утраченных значений дискретных последовательностей сигналов используют специальные методы частотно-временного преобразования или разложения во временные ряды [29 – 37]. В первом случае в основе методов восстановления лежит реконструкция формы сигнала с применением различных адаптивных ядер, представляющих собой функции с изменяющимися параметрами. Такой подход улучшает частотное разрешение, но может оказать искажающее влияние на частотные характеристики анализируемого сигнала. Во втором случае методы восстановления обеспечивают получение оценок значений отсчетов непосредственно во временной области на основе аппроксимации утраченного фрагмента последовательности с помощью простых функций или полиномов. В частности, эффективным является метод локальной аппроксимации. Согласно этому методу для каждого проблемного участка в его окрестности выбирается подпоследовательность с известными значениями

отсчетов, по которой осуществляется построение аппроксимирующей модели. При этом восстановление может осуществляться как по одной подпоследовательности, находящейся перед или после восстанавливаемого участка, так и одновременной по обоим [38, 39]. Преимущество применения процедуры аппроксимации заключается в том, что оценка утраченных значений ведется с учетом изменения значений отсчетов подпоследовательностей, находящихся в непосредственной близости к восстанавливаемому участку. Однако локальная аппроксимация применима только на коротких временных интервалах, когда можно считать, что между значениями проблемных отсчетов и отсчетов по которым осуществляется восстановление имеет место устойчивая зависимость. Кроме того, дискретная последовательность может иметь несколько проблемных участков, находящихся достаточно близко по отношению друг к другу. В этом случае для некоторых восстанавливаемых участков может оказаться проблематичным выбор подпоследовательностей в непосредственной близости от них с необходимым числом отсчетов для выполнения алгоритма их восстановления.

Таким образом, актуальной задачей является уменьшение влияния неопределенности изменения сигнала во времени на получаемые оценки значений утраченных отсчетов, и, следовательно, на результаты цифровой обработки сигнала. При этом математическое решение этой задачи должно способствовать разработке эффективного в вычислительном отношении алгоритмического обеспечения и соответствующего ему программного обеспечения, которое можно использовать в асинхронном режиме восстановления участка с проблемными отсчетами непосредственно в процессе цифровой обработки сигнала.

**2. Постановка задачи.** Восстановление проблемных участков дискретной последовательности сигнала осуществляется по имеющейся выборке его отсчетов с известными подтвержденными значениями. Фактически задача восстановления сводится к нахождению математических зависимостей, позволяющих вычислительным путем получить численные оценки значений утраченных отсчетов.

С математической точки зрения отсчет сигнала, зафиксированный в определенный момент времени, представляет собой одно из возможных значений, которое он может принимать в пределах допустимой области его изменения. При этом в реальном мире сигналы не являются абсолютно произвольными функциями времени, и их значения не могут изменяться произвольным образом.

Сигналы являются носителями некоторой информации, и изменение их значений во времени имеет присущие им характерные особенности. Учет таких особенностей позволяет спроектировать эффективный алгоритм восстановления дискретной последовательности сигнала. В частности, это касается корреляционной зависимости между значениями сигнала. Она позволяет получить количественную оценку степени статистической связи значений сигнала во времени в процессе его распространения.

Задачу восстановления отсчетов дискретной последовательности сигнала будем решать исходя из того, что он удовлетворяет условиям стационарности или стационарен хотя бы в широком смысле (квазистационарный). Выполнение последнего условия позволяет обобщить возможность применения разрабатываемого алгоритма восстановления отсчетов также и на те сигналы, статистические характеристики которых можно считать неизменными в пределах интервала времени формирования их дискретных последовательностей. Прежде всего, это касается постоянства корреляционной зависимости значений сигнала при его распространении во времени. Также будем считать, что частотный состав сигнала находится в пределах диапазона с верхней граничной частотой  $F_{\max}$ .

Пусть в процессе цифрового анализа сигнал  $x(t)$ , удовлетворяющего выше приведенным условиям, осуществлена его равномерная дискретизация с частотой  $F_s \geq 2F_{\max}$ . В результате этого в моменты времени  $t_n = n\Delta t$  будет сформирована дискретная последовательность отсчетов  $x_n = x(n\Delta t)$ , где  $\Delta t = 1/F_s$  и  $n = 0, 1, 2, 3, \dots, N-1$ . Математически такую последовательность можно представить в виде  $N$ -мерного вектора  $\mathbf{x} = (x_0, x_1, x_2, x_2, \dots, x_{N-1})$ .

Допустим, что значения отсчетов  $x_n$  с индексами  $n \in [0; k]$  и  $n \in [k + M + 1; N - 1]$  известны точно и их положение соответствует равномерной временной сетке с заданной частотой дискретизации  $F_s$ . Значения  $M$  отсчетов  $x_n$  с индексами  $n \in [k + 1; k + M]$  образуют проблемный участок последовательности. Они могут быть утрачены, искажены или иметь смещение по отношению к равномерной временной сетке. На рисунке 1 схематично изображена дискретная последовательность с проблемным участком.

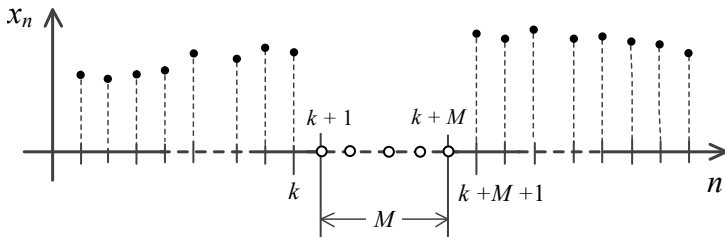


Рис. 1. Дискретная последовательность с проблемным участком

Как было отмечено выше одной из важнейших статистических характеристик сигнала является корреляционная зависимость динамики изменения его значений во времени. Для получения представления о такой зависимости используется корреляционная функция (КФ). Она характеризует линейную стохастическую связь между значениями сигнала. КФ равномерной дискретной последовательности  $x_n = x(n\Delta t)$  является функцией двух аргументов времени  $t_i = i\Delta t$  и  $t_j = j\Delta t$  :

$$R_{XX}(t_i, t_j) = M[x(i\Delta t) x(j\Delta t)] = M[x_i^o x_j^o], \quad (1)$$

где  $x_i^o = (x_i - m_x)$  и  $x_j^o = (x_j - m_x)$  – центрированные значения отсчетов;  $m_x$  – математическое ожидание сигнала  $x(t)$  ;  $M[...]$  – оператор оценки математического ожидания.

КФ является симметричной. Если сигнал стационарен хотя бы в широком смысле, то она является функцией только разности ее аргументов, характеризующих интервал сдвига отсчетов во времени:

$$R_{XX}(t_i, t_j) = R_{XX}(t_j, t_i) = R_{XX}(t_j - t_i) = R_{XX}(t_i - t_j) = R_{j-i} = R_{i-j}. \quad (2)$$

На практике оценка КФ анализируемого сигнала  $x(t)$  может быть получена по выборке его отсчетов с известными значениями. При этом в качестве математического ожидания  $m_x$  сигнала  $x(t)$  принимается оценка среднего значения отсчетов, вычисленная по той же самой выборке, по которой вычисляется и оценка КФ. Однако в процессе таких вычислений следует учитывать статистическую погрешность, обусловленную конечным числом отсчетов выборки, т.е.

ее объем должен быть достаточным. Поэтому применение алгоритма восстановления отсчетов дискретной последовательности на основе учета их корреляционной связи должно принимать во внимание данное обстоятельство. Фактически оно определяет границы области возможного использования соответствующего алгоритма, в пределах которых восстановление значений отсчетов будет численно стабильным и обеспечивать получение удовлетворительных результатов.

Принимая во внимание корреляционную зависимость во времени между значениями сигнала, задачу восстановления проблемного участка дискретной последовательности его отсчетов будем решать с использованием метода прямого и обратного прогнозирования. По отношению к физическим сигналам задача прямого прогнозирования заключается в предсказании его развития и оценке значений на последующем участке в зависимости от предыдущего. В свою очередь задача обратного прогнозирования предполагает по имеющимся отсчетам оценить значения отсчетов на предыдущем участке развития сигнала.

**3. Восстановление значений отсчетов дискретной временной последовательности сигнала на основе прямого прогнозирования.** Рассмотрим модель порядка  $p$ , которая выражает значения последующих отсчетов дискретной последовательности в виде линейной комбинации конечного числа предыдущих значений [40, 41]:

$$\hat{x}_n = \sum_{j=1}^p a_j x_{n-j} . \quad (3)$$

где  $a_j$  – коэффициенты модели.

Из (3) следует, что если значения отсчетов последовательности  $x_n$  известны для  $n = i-1, i-2, \dots, i-p$ , то можно вычислить значения последующих отсчетов этой последовательности для  $n = i, i+1, i+2, \dots$ . С математической точки зрения модель данного вида позволяет рекуррентно осуществлять вычисление оценок значений утраченных отсчетов  $x_n$  для  $n = k+1, k+2, \dots, k+M$ , т.е. определяет вычислительные процедуры прямого прогнозирования (прогнозирования вперед).

Совершив последовательно  $i$  рекурсий и тем самым осуществив прогноз на  $i$  шагов вперед, получаем:

$$\hat{x}_{n+i} = \sum_{j=1}^p \alpha_{i,j} x_{n-j}, \quad n = k+1, \quad i = 1, 2, \dots, M-1, \quad (4)$$

где

$$\alpha_{i,j} = \sum_{\vartheta=1}^i w_{\vartheta} a_{j+i-\vartheta}, \quad (5)$$

$$a_0 = 1, \quad a_n = 0 \quad \text{при } n > p;$$

$$w_{\vartheta} = \sum_{j=1}^p a_j w_{\vartheta-j}, \quad \vartheta = 1, 2, 3, \dots, i; \quad (6)$$

$$w_0 = 1, \quad w_{-\vartheta} = 0.$$

С учетом того, что в (4)  $n = k+1$ , будем иметь:

$$\hat{x}_{k+1+i} = \sum_{j=1}^p \alpha_{i,j} x_{k+1-j}, \quad i = 1, 2, \dots, M-1. \quad (7)$$

Введем обозначение  $m = 1 + i$ . Тогда получаем математическое соотношение для восстановления значений отсчетов:

$$\hat{x}_{k+m} = \sum_{j=1}^p \alpha_{m-1,j} x_{k+1-j}, \quad m = 1, 2, \dots, M. \quad (8)$$

Соотношение (8) фактически представляет собой модель скользящего среднего, где  $\alpha_{m-1,j}$  по своей сути являются весовыми коэффициентами [40, 41]. Согласно этой модели вычисление оценок значений восстанавливаемых отсчетов  $x_n$  для  $n = k+1, k+2, \dots, k+M$ , осуществляется по конечной выборке предыдущих отсчетов с известными значениями  $x_n$ , где  $n \in [k+1-p; k]$ . Принципиально важным является то, что восстановление значения любого из этих отсчетов осуществляется по отсчетам одной и той же выборке. При этом коэффициенты  $\alpha_{m-1,j}$  определяют индивидуальный вес каждого отсчета этой выборки при оценке значения восстанавливаемого отсчета с индексом  $k+m$ , что эквивалентно прямому прогнозированию на  $m$  шагов.

Значения весовых коэффициентов  $\alpha_{m-1,j}$  будем вычислять исходя из критерия минимума среднеквадратической погрешности:

$$\delta_{p,m} = M[(\hat{x}_{k+m} - x_{k+m})^2] \rightarrow \min, \quad (9)$$

где  $M[\dots]$  – оператор оценки математического ожидания.

Для обеспечения выполнения критерия (9) найдем частные производные первого порядка по каждому из коэффициентов  $\alpha_{m-1,j}$ , полагая при этом остальные коэффициенты постоянными, и приравняем эти производные нулю. Тогда получаем систему уравнений:

$$\frac{\partial \delta_{p,m}}{\partial \alpha_{m-1,q}} = 2 \sum_{j=1}^p \alpha_{m-1,j} M[x_{k+1-j} x_{k+1-q}] - 2M[x_{k+m} x_{k+1-q}] = 0, \quad (10)$$

$$q = 1, 2, 3, \dots, p.$$

Будем считать, что восстанавливаемая последовательность имеет нулевое математическое ожидание, т.е. значения ее отсчетов являются центрированными. В противном случае по отсчетам с известными значениями всегда можно вычислить их среднее значение, которое принимается в качестве оценки математического ожидания, и осуществить относительно него операцию центрирования. Принимая это во внимание с учетом (1) и (2) в (10) будем иметь:

$$M[x_{k+1-j} x_{k+1-q}] = R_{q-j}, \quad M[x_{k+m} x_{k+1-q}] = R_{1-m-q} = R_{m+q-1}. \quad (11)$$

В результате получаем систему линейных уравнений, решение которой позволяет получить соотношения для вычисления значений весовых коэффициентов  $\alpha_{m-1,j}$ :

$$\sum_{j=1}^p \alpha_{m-1,j} R_{q-j} = R_{q+m-1}, \quad q = 1, 2, 3, \dots, p, \quad m = 1, 2, \dots, M. \quad (12)$$

В матричной форме система уравнений (12) будет иметь вид:



$$\begin{pmatrix} R_0 & R_{-1} & \dots & R_{1-p} \\ R_1 & R_0 & \dots & R_{2-p} \\ \dots & \dots & \dots & \dots \\ R_{p-1} & R_{p-2} & \dots & R_0 \end{pmatrix} \times \begin{pmatrix} \alpha_{m-1,1} \\ \alpha_{m-1,2} \\ \dots \\ \alpha_{m-1,p} \end{pmatrix} = \begin{pmatrix} R_m \\ R_{1+m} \\ \dots \\ R_{p+m-1} \end{pmatrix}, \quad (13)$$

$$m = 1, 2, \dots, M.$$

Из (13) следует, что для определения значений  $\alpha_{m-1,j}$  необходимо иметь множество отсчетов КФ  $\{R_0, R_1, \dots, R_{p+m-1}\}$  восстанавливаемой последовательности анализируемого сигнала. Вследствие того, что восстановлению подлежат  $M$  значений отсчетов, то всего необходимо иметь  $p + M - 1$  отсчетов КФ  $\{R_0, R_1, \dots, R_{p+M-1}\}$ . При этом для любого  $m \in [1; M]$  в ходе определения значений  $\alpha_{m-1,j}$  основная матрица системы уравнений (13), состоящая из отсчетов КФ  $R_{q-j}$ , остается неизменной. В зависимости от порядкового номера восстанавливаемого отсчета  $m$  будет меняться лишь только столбец свободных членов, состоящий из отсчетов КФ  $R_{q+m-1}$ . Для стационарного сигнала оценки отсчетов КФ  $\{R_0, R_1, \dots, R_{p+M-1}\}$ , необходимые для определения значений  $\alpha_{m-1,j}$ , могут быть вычислены по известным отсчетам последовательности на любом ее участке независимо от восстанавливаемого участка.

С целью упрощения математических записей при решении системы уравнений (12) введем следующие обозначения:

$$\alpha_{m-1,j} = A_{m,j}, \quad \lambda_{q,j} = R_{q-j}, \quad B_{q,m} = R_{q+m-1}. \quad (14)$$

Тогда система уравнений (12) примет вид:

$$\sum_{j=1}^p A_{m,j} \lambda_{q,j} = B_{q,m}, \quad q = 1, 2, 3, \dots, p, \quad m = 1, 2, \dots, M. \quad (15)$$

В матричной форме для системы уравнений (15) будем иметь:

$$\begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} & \dots & \lambda_{1,p} \\ \lambda_{2,1} & \lambda_{2,2} & \dots & \lambda_{2,p} \\ \dots & \dots & \dots & \dots \\ \lambda_{p,1} & \lambda_{p,2} & \dots & \lambda_{p,p} \end{pmatrix} \times \begin{pmatrix} A_{m,1} \\ A_{m,2} \\ \dots \\ A_{m,p} \end{pmatrix} = \begin{pmatrix} B_{1,m} \\ B_{2,m} \\ \dots \\ B_{p,m} \end{pmatrix}, \quad (16)$$

$$m = 1, 2, \dots, M.$$

В результате решения системы уравнений (16) методом Гаусса получаем математическое соотношение для вычисления оценок искомых весовых коэффициентов:

$$\alpha_{m-1,j} = A_{m,j} = \frac{1}{\lambda_{j,j}^j} \left( B_{j,m}^{(j)} - \sum_{i=j+1}^p \lambda_{j,i}^j \alpha_{m-1,i} \right), \quad j = 1, 2, 3, \dots, p; \quad (17)$$

где

$$\lambda_{q,j}^{(i+1)} = \lambda_{q,j}^{(i)} - D_{q,i} \lambda_{i,j}^{(i)}; \quad (18)$$

$$B_{q,m}^{(i+1)} = B_{q,m}^{(i)} - D_{q,i} B_{i,m}^{(i)}; \quad (19)$$

$$D_{q,i} = \lambda_{q,i}^{(i)} / \lambda_{i,i}^i; \quad (20)$$

$$i = 1, 2, 3, \dots, p-1; \quad q = i+1, i+2, \dots, p; \quad j = i+1, i+2, \dots, p.$$

Отметим, что

$$\lambda_{1,j}^{(1)} = R_{1-j}, \quad j = 1, 2, 3, \dots, p; \quad (21)$$

$$B_{1,m}^{(1)} = R_m, \quad m = 1, 2, \dots, M. \quad (22)$$

Среднеквадратическая погрешность (9) вычисления оценки утраченного значения отсчета  $\hat{x}_{k+m}$  согласно (8) будет равна:

$$\delta_{p,m} = M \left[ \left( \sum_{j=1}^p \alpha_{m-1,j} x_{k+1-j} - x_{k+m} \right)^2 \right]. \quad (23)$$

После возведения в квадрат математического соотношения в круглых скобках и приняв во внимание, что операция математического ожидания является линейной операцией, получаем:

$$\delta_{p,m} = \sum_{j=1}^p \alpha_{m-1,j} \sum_{i=1}^p \alpha_{m-1,i} M[x_{k+1-j}x_{k+1-i}] - 2 \sum_{j=1}^p \alpha_{m-1,j} M[x_{k+1-j}x_{k+m}] - M[x_{k+m}^2]. \quad (24)$$

В (24) для математических ожиданий произведений значений отсчетов будем иметь:

$$M[x_{k+1-j}x_{k+1-i}] = R_{j-i}, \quad M[x_{k+1-j}x_{k+m}] = R_{m+j-1}, \quad M[x_{k+m}^2] = R_0. \quad (25)$$

Отметим, что  $R_0 = \sigma_x^2$ , где  $\sigma_x^2$  дисперсия анализируемого сигнала, а  $\sigma_x$  среднеквадратическое отклонение.

С учетом (25) получаем:

$$\delta_{p,m} = \sum_{j=1}^p \alpha_{m-1,j} \left( \sum_{i=1}^p \alpha_{m-1,i} R_{j-i} - 2R_{m+j-1} \right) - R_0. \quad (26)$$

Принимая во внимание (12), окончательно получаем, что среднеквадратическая погрешность вычисления  $\hat{x}_{k+m}$  будет равна:

$$\delta_{p,m} = R_0 - \sum_{j=1}^p \alpha_{m-1,j} R_{j+m-1}, \quad m = 1, 2, \dots, M. \quad (27)$$

В качестве критерия выбора порядка модели  $p$  будем использовать среднее значение среднеквадратической погрешности (27) нормированное относительно  $R_0$ :

$$\varepsilon_p = \frac{1}{M} \sum_{m=1}^M \frac{\delta_{p,m}}{R_0} \alpha_{m-1,j} = 1 - \frac{1}{M} \sum_{j=1}^p \sum_{m=0}^{M-1} \alpha_{m-1,j} \rho_{j+m-1}, \quad (28)$$

$$m = 1, 2, \dots, M.$$

где  $\rho_{j+m} = R_{j+m} / R_0$  – нормированные значения отсчетов КФ.

**4. Восстановление значений отсчетов дискретной временной последовательности сигнала на основе обратного прогнозирования.** Соотношение (8), определяющее процедуру восстановления значений отсчетов на основе прямого прогнозирования, может быть непосредственно применено для восстановления значений отсчетов на основе обратного прогнозирования. Это объясняется тем, что оно получено с учетом корреляционной зависимости значений сигнала при его распространении во времени. Весовые коэффициенты  $\alpha_{m-1,j}$  в составе соотношения (8) вычисляются исходя из оценки отсчетов КФ. Они определяют индивидуальный вес каждого отсчета  $x_n$  выборки с индексами  $n = i - 1, i - 2, \dots, i - p$ , по которой осуществляется восстановление значения отсчетов  $x_n$  с индексом  $n = k + m$  по мере увеличения их порядкового номера  $m = 1, 2, 3, \dots, M$ . Как было отмечено выше КФ является симметричной функцией (2). Поэтому, если известны значения отсчетов с индексами  $n = k + M + 1, k + M + 2, \dots, k + M + p$ , то (8) позволяет вычислять оценки утраченных значений отсчетов  $x_n$  с индексами  $n = k + m$  в обратном порядке для  $m = M, M - 1, M - 2, \dots, 3, 2, 1$ . В соответствии с этим будем иметь:

$$\hat{x}_{k+M+1-m} = \sum_{j=1}^p \alpha_{m-1,j} x_{k+M+j}, \quad m = 1, 2, \dots, M. \quad (29)$$

Введем обозначение  $i = M + 1 - m$  и сделаем тождественную замену  $m \equiv i$ . Тогда, сохранив прямой порядок вычисления оценок значений отсчетов, получаем:

$$\hat{x}_{k+m} = \sum_{j=1}^p \alpha_{M-m,j} x_{k+M+j}, \quad m = 1, 2, \dots, M. \quad (30)$$

Согласно (30) процедура восстановления значений отсчетов осуществляется также как и в (8) по  $p$  значениям отсчетов одной и той же выборки, но находящейся после восстанавливаемого участка. При этом весовые коэффициенты  $\alpha_{M-m,j}$  используются в обратном

порядке с увеличением индекса  $m$ , что эквивалентно осуществлению прямого прогноза на  $M - m$  шагов назад. Отсюда следует, что среднеквадратическая погрешность оценок значений отсчетов (30) будет определяться согласно (27). В свою очередь в качестве критерия выбора порядка модели  $p$  также будем использовать нормированное среднее значение среднеквадратической погрешности (28).

Важно отметить, что для соотношений (8) и (30), определяющих вычислительные процедуры восстановления значений отсчетов при прямом и обратном прогнозировании, метрологические характеристики идентичны. Вследствие этого выбор процедуры восстановления значений отсчетов на основе прямого или обратного прогнозирования будет определяться только достаточностью необходимого числа отсчетов выборки с известными значениями до или после проблемного участка, по которым осуществляется процедура восстановления.

**5. Восстановление значений отсчетов дискретной временной последовательности сигнала одновременно на основе прямого и обратного прогнозирования.** В простейшем случае задача восстановления утраченных значений отсчетов дискретной последовательности на основе прямого (8) и обратного прогнозирования (30) может быть решена следующим образом:

$$\left\{ \begin{array}{l} \hat{x}_{k+m} = \sum_{j=1}^p \alpha_{m-1,j} x_{k+1-j}, \quad m = 1, 2, 3, \dots, \eta; \\ \hat{x}_{k+m} = \sum_{j=1}^p \alpha_{M-m,j} x_{k+M+j}, \quad m = \eta + 1, \eta + 2, \dots, M; \end{array} \right. \quad (31)$$

где число  $\eta$  представляет собой целую часть отношения  $M / 2$ .

В данном случае в два раза уменьшается длительность прогнозирования как вперед, так и назад. Вследствие этого улучшаются метрологические характеристики результатов восстановления значений отсчетов. Данный подход к процедуре восстановления значений отсчетов не представляет особой сложности. Это объясняется тем, что необходимые для вычисления значений весовых коэффициентов  $\alpha_{m-1,j}$  оценки КФ для стационарных или квазистационарных сигналов могут быть вычислены по любой выборке отсчетов с известными значениями восстанавливаемой последовательности независимо от того какой вид прогнозирования осуществляется. Однако следует помнить, что число отсчетов такой

выборки должно быть достаточным для вычисления необходимого числа оценок КФ с учетом их статистической погрешности. При этом все-таки желательно, чтобы выборка была бы близко расположенной по отношению к восстанавливаемому участку.

Далее решение задачи по восстановлению утраченных значений отсчетов последовательности рассмотрим на основе совместного осуществления прямого и обратного прогнозирования. За основу примем соотношения (8) и (30). Тогда, сделаем в (30) формальную замену вида  $\alpha_{M-m,j} = \beta_{M-m,j}$ , процедуру восстановления значений отсчетов будем осуществлять следующим образом:

$$\hat{x}_{k+m} = \sum_{j=1}^p \alpha_{m-1,j} x_{k+1-j} + \sum_{j=1}^p \beta_{M-m,j} x_{k+M+j}, \quad m = 1, 2, \dots, M. \quad (32)$$

В (32) весовые коэффициенты  $\alpha_{m-1,j}$  и  $\beta_{M-m,j}$  будем находить также исходя из критерия минимума среднеквадратической погрешности (9). В соответствии с этим найдем частные производные первого порядка по каждому из этих коэффициентов, полагая все остальные коэффициенты постоянными. Приравняв полученные производные нулю с учетом (11) будем иметь систему уравнений:

$$\begin{cases} \sum_{j=1}^p \alpha_{m-1,j} R_{q-j} + \sum_{j=1}^p \beta_{M-m,j} R_{M-1+j+q} = R_{m-1+q}; \\ \sum_{j=1}^p \alpha_{m-1,j} R_{M-1+j+q} + \sum_{j=1}^p \beta_{M-m,j} R_{q-j} = R_{M-m+q}; \end{cases} \quad (33)$$

$$q = 1, 2, 3, \dots, p, \quad m = 1, 2, \dots, M.$$

Если в системе (33) изменение индекса  $m$  рассматривать в обратном порядке, т.е. когда  $m = M, M-1, M-2, \dots, 3, 2, 1$ , то первое уравнение в этой системе примет вид:

$$\sum_{j=1}^p \alpha_{M-m,j} R_{q-j} + \sum_{j=1}^p \beta_{m-1,j} R_{M-1+j+q} = R_{M-m+q}. \quad (34)$$

Сопоставляя (34) со вторым уравнением в (33), приходим к выводу о справедливости равенств  $\beta_{m-1,j} = \alpha_{m-1,j}$  и  $\beta_{M-m,j} = \alpha_{M-m,j}$ . Тогда соотношение (32) можно записать в следующем виде:

$$\hat{x}_{k+m} = \sum_{j=1}^p \alpha_{m-1,j} x_{k+1-j} + \sum_{j=1}^p \alpha_{M-m,j} x_{k+M+j}, \quad m = 1, 2, \dots, M. \quad (35)$$

В свою очередь система уравнений (33) примет вид:

$$\begin{cases} \sum_{j=1}^p \alpha_{m-1,j} R_{q-j} + \sum_{j=1}^p \alpha_{M-m,j} R_{M-1+j+q} = R_{m-1+q}; \\ \sum_{j=1}^p \alpha_{m-1,j} R_{M-1+j+q} + \sum_{j=1}^p \alpha_{M-m,j} R_{q-j} = R_{M-m+q}; \end{cases} \quad (36)$$

$$q = 1, 2, 3, \dots, p, \quad m = 1, 2, \dots, M.$$

Найдем решение системы уравнений (36). Во-первых, сложим верхнее и нижнее уравнения, образующие эту систему. Во-вторых, вычтем из верхнего уравнения нижнее. Соответственно получим:

$$\sum_{j=1}^p (\alpha_{m-1,j} + \alpha_{M-m,j})(R_{q-j} + R_{M-1+j+q}) = R_{m-1+q} + R_{M-m+q}, \quad (37)$$

$$\sum_{j=1}^p (\alpha_{m-1,j} - \alpha_{M-m,j})(R_{q-j} - R_{M-1+j+q}) = R_{m-1+q} - R_{M-m+q}. \quad (38)$$

В (37) и (38)  $q = 1, 2, 3, \dots, p$  и  $m = 1, 2, \dots, M$ . Структура этих уравнений соответствует уравнению (15). При этом для (37) будем иметь:

$$\begin{aligned} A_{m,j} &= \alpha_{m-1,j} + \alpha_{M-m,j}, \quad \lambda_{q,j} = R_{q-j} + R_{M-1+j+q}, \\ B_{q,m} &= R_{m-1+q} + R_{M-m+q}. \\ \lambda_{1,j}^{(1)} &= R_{1-j} + R_{M+j}, \quad j = 1, 2, 3, \dots, p. \\ B_{1,m}^{(1)} &= R_m + R_{M-m+1}, \quad m = 1, 2, \dots, M. \end{aligned} \quad (39)$$

В свою очередь для (38) будем иметь:

$$\begin{aligned} A_{m,j} &= \alpha_{m-1,j} - \alpha_{M-m,j}, \quad \lambda_{q,j} = R_{q-j} - R_{M-1+j+q}, \\ B_{q,m} &= R_{m-1+q} - R_{M-m+q}. \end{aligned} \quad (40)$$

$$\lambda_{1,j}^{(1)} = R_{1-j} - R_{M+j}, \quad j = 1, 2, 3, \dots, p.$$

$$B_{1,m}^{(1)} = R_m - R_{M-m+1}, \quad m = 1, 2, \dots, M.$$

С учетом введенных обозначений (39) и (40) решение каждой из систем уравнений (37) и (38) определяется соотношениями (17)-(22).

В результате решения системы (37) получаем:

$$A_{m,j} = (\alpha_{m-1,j} + \alpha_{M-m,j}) = W_1(m, j), \quad j = 1, 2, 3, \dots, p. \quad (41)$$

Соответственно в результате решения системы (38) получаем:

$$A_{m,j} = (\alpha_{m-1,j} - \alpha_{M-m,j}) = W_2(m, j), \quad j = 1, 2, 3, \dots, p. \quad (42)$$

Окончательно получаем, что весовые коэффициенты равны:

$$\alpha_{m-1,j} = \frac{W_1(m, j) + W_2(m, j)}{2}, \quad j = 1, 2, 3, \dots, p. \quad (43)$$

При вычислении оценки утраченного значения отсчета  $\hat{x}_{k+m}$  согласно (35) среднеквадратическая погрешность будет равна:

$$\begin{aligned} \delta_{p,m} &= \text{M}[(\sum_{j=1}^p (\alpha_{m-1,j} x_{k+1-j} + \alpha_{M-m,j} x_{k+M+j}) - x_{k+m})^2] = \\ &= R_0 - \sum_{j=1}^p (\alpha_{m-1,j} R_{m-1+j} + \alpha_{M-m,j} R_{M-m+j}). \end{aligned} \quad (44)$$

Как и ранее в качестве критерия выбора порядка модели  $p$  будем использовать среднее значение среднеквадратической погрешности нормированное относительно  $R_0$ :

$$\varepsilon_{p,m} = \frac{1}{M} \sum_{m=1}^M \frac{\delta_{p,m}}{R_0} = 1 - \frac{2}{M} \sum_{j=1}^p \sum_{m=0}^{M-1} \alpha_{m,j} \frac{R_{m+j}}{R_0}, \quad m = 1, 2, \dots, M. \quad (45)$$

**4. Численные эксперименты по восстановлению дискретных последовательностей сигнала.** На основе соотношений (8), (30), (35) с учетом полученных решений для вычисления весовых



коэффициентов (17) и (41)-(43) осуществлена разработка алгоритмического обеспечения для восстановления утраченных значений отсчетов дискретной временной последовательности сигнала в процессе построения моделей скользящего среднего при прямом, обратном и совместном прогнозировании [42]. Данное алгоритмическое обеспечение реализовано в виде функционально завершенного метрологически значимого программного модуля. Разработка модуля осуществлена в соответствии с обязательными требованиями нормативных документов, регламентирующих обеспечение единства процедур измерения и вычислительных процедур, которые влияют на метрологические характеристики результатов измерений [43, 44]. Модуль предназначен для работы в режиме асинхронного управления с возможностью выбора одного из трех альтернативных вариантов процедуры восстановления значений отсчетов. Такой режим работы позволяет осуществлять модулю свои функции без прерывания выполнения основной прикладной программы в процессе обработки дискретной последовательности сигнала. Предусмотрена возможность многократного использования модуля в составе метрологически значимого программного обеспечения для multifunctional систем анализа сложных сигналов [45 – 47].

Тестирование алгоритмического обеспечения и проверка работоспособности программного модуля по выполнению функциональных возможностей восстановления утраченных значений отсчетов осуществлялись на основе имитационного моделирования дискретной последовательности сигнала [48]. Модель сигнала представляла собой аддитивную смесь, содержащую  $L$  независимых гармонических компонент со случайными начальными фазами  $\varphi_v \in [-\pi; +\pi]$ , которые задавались в соответствии с равномерным законом распределения. Значения амплитуд  $A_v$  и частот  $f_v$  гармонических компонент задавались в пределах от нуля до единицы и интерпретировались как нормированные соответственно по отношению к значению наибольшей амплитуде гармоники, присутствующей в составе модели сигнала, и к значению верхней границы частотного диапазона, занимаемого гармониками. При этом постоянная составляющая модели принималась равной нулю. Интерпретация значений амплитуд  $A_v$  и частот  $f_v$  гармонических компонент как номерованных позволила в ходе проведения тестовых испытаний формализовать процедуру исследования метрологических возможностей разработанных алгоритмов для восстановления

значений отсчетов сигналов по отношению к их амплитудно-частотным характеристикам. Использование нормированных частот объясняется также тем, что в процессе восстановления значений отсчетов важны не абсолютные значения частот гармоник, а их соотношение в составе сигнала. Кроме того, такой подход позволил достаточно просто контролировать изменение во времени формы модели сигнала и его КФ. Отметим, что для гармонической компоненты с амплитудой  $A_v$ , частотой  $f_v$  и начальной фазой  $\varphi_v$  КФ будет косинусоидой с амплитудой  $A_v^2/2$  и той же самой частотой:

$$R_{XX}(\tau) = \frac{A_v^2}{2} \cos 2\pi f_v \tau. \quad (46)$$

КФ модели сигнала, являющегося суммой  $L$  независимых гармонических компонент, будет представлять собой сумму их КФ:

$$R_{XX}(\tau) = \frac{1}{2} \sum_{v=1}^L A_v^2 \cos 2\pi f_v \tau. \quad (47)$$

Следует отметить так же то, что КФ гармонического сигнала не сохраняет информацию о его начальной фазе. Имея только оценки значений КФ можно получить бесконечный ансамбль гармоник с одинаковыми по форме КФ, но с отличающимися фазовыми сдвигами во времени по отношению друг к другу. Это означает, что нельзя однозначно восстановить гармонический сигнал или сигнал, содержащий гармонические компоненты, только по значениям оценок его КФ. Еще раз отметим, что КФ позволяет получить представление только о линейной стохастической связи между значениями сигнала в различные моменты времени. Поэтому использование в качестве модели сигнала суммы гармонических компонент со случайными начальными фазами позволяет оценить, насколько точно разработанное алгоритмическое обеспечение по имеющимся значениям отсчетов только на основе оценки их корреляционной связи позволяет осуществить прямое и обратное прогнозирование во времени и восстановить значения отсчетов на проблемных участках дискретной последовательности сигнала.

Тестовые испытания главным образом были направлены на оценку результатов восстановления значений отсчетов в зависимости от выбора порядка модели восстановления. При этом осуществлялась предварительная оценка значений КФ с необходимой точностью

в соответствии с принципами корреляционного анализа сигналов. Как было отмечено выше, для стационарного сигнала оценка КФ может быть вычислена на любом участке с достоверными значениями отсчетов в пределах обрабатываемой последовательности. Поэтому вычисление оценок КФ может быть осуществлено только один раз, даже если необходимо восстановить несколько участков с проблемными отсчетами. Хотя следует иметь в виду, что по возможности участок последовательности, по отсчетам которого будут вычисляться оценки КФ, должен находиться как можно ближе к восстанавливаемому участку.

В процессе тестовых экспериментов формирование дискретной последовательности реализации модели сигнала осуществлялось как с повышенной частотой дискретизации, так и в случае, когда частоты гармонических компонент были близки к частоте дискретизации, задаваемой согласно теореме Котельникова (Найквиста). В первом случае, предоставляется возможность исследовать характерные особенности алгоритмов, связанные с «чувствительностью» осуществлять прогнозирование и восстановление значений соседних отсчетов с мало отличающимися значениями, когда корреляционная зависимость между ними также изменяется незначительно. Во втором случае предоставляется возможность исследовать способность алгоритмов прогнозировать и восстанавливать значения отсчетов по мере увеличения расстояния и ослабевании корреляционной зависимости между ними.

Одна из реализаций модели сигнала содержала пять гармонических компонент, параметры которых приведены в таблице 1.

Таблица 1. Параметры гармонических компонент

$v$	$A_v$	$f_v$	$\varphi_v$
1	0,15	0,2	0,959222
2	0,25	0,35	-1,559470
3	1,0	0,5	1,275801
4	0,5	0,75	-1,444726
5	0,1	1,0	1,314656

Для гармонического состава модели сигнала, приведенного в таблице 1, компонентой с наибольшей нормированной частотой является синусоида под номером пять, частота которой равна единице. Исходя из этого, формирование дискретной последовательности с повышенной частотой дискретизации, в частности, осуществлялось

с нормированной частотой  $F_s = 100f_s = 100$ , т.е. интервал дискретизации был равен  $\Delta t = 0,01$ . В процессе экспериментальных исследований конечное число отсчетов на любом участке последовательности можно интерпретировать как отсчеты с утраченными значениями. Так, например, в качестве проблемного участка рассматривались отсчеты с индексами  $n \in [1201; 1215]$ , т.е.  $k = 1200$  и  $M = 15$ . На рисунках 2 и 3 приведены соответственно фрагмент реализации модели сигнала с учетом места нахождения проблемного участка и сам проблемный участок последовательности. На рисунке 4 приведен фрагмент оценки КФ, вычисленной по отсчетам данной последовательности.

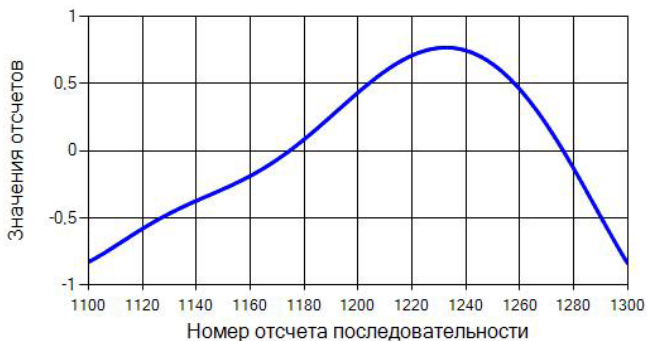


Рис. 2. Фрагмент реализации модели сигнала с проблемным участком

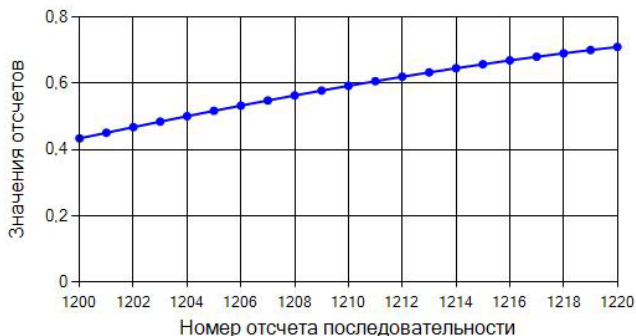


Рис. 3. Проблемный участок последовательности;  $F_s = 100f_s = 100$

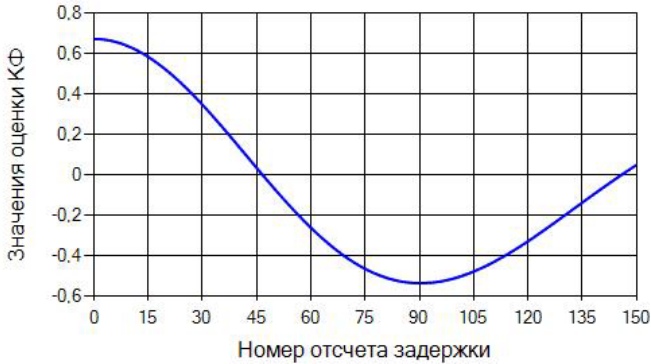


Рис. 4. Фрагмент оценки КФ;  $F_s = 100f_5 = 100$

В таблицах 2, 3 и 4 представлены результаты тестовых экспериментов по восстановлению проблемного участка последовательности для  $n \in [1201; 1215]$  при прямом, обратном и совместном прогнозировании. Порядок модели восстановления  $p$  был равен пяти, десяти и пятнадцати. С целью количественной оценки отклонения восстановленных числовых значений отсчетов от их истинных значений вычислялась относительная погрешность  $\delta_{k+m}^x = (\hat{x}_{k+m} - x_{k+m}) / x_{k+m}$ . В нижней части таблиц 2, 3 и 4 также приведены значения для среднеквадратической погрешности

$$\delta_{\text{MSE}}^x = \sqrt{\frac{1}{M} \sum_{m=1}^M (\hat{x}_{k+m} - x_{k+m})^2}$$
 и средней абсолютной погрешности

$$\delta_{\text{MAE}}^x = \frac{1}{M} \sum_{m=1}^M |\hat{x}_{k+m} - x_{k+m}|.$$

Как следует из таблиц 2, 3 и 4 результаты тестовых экспериментов не содержат грубых систематических ошибок, которые существенно бы искажали истинные значения восстанавливаемых отсчетов. С увеличением порядка модели  $p$  относительная погрешность уменьшается. При этом уже при порядке модели  $p = 10$  в пределах всего проблемного участка, на котором было осуществлено восстановление отсчетов, изменение относительной погрешности для любой из рассматриваемых процедур восстановления не превышает по абсолютной величине значения 0,01, т.е. одного процента от истинных значений отсчетов. Среднеквадратическая и средняя абсолютная погрешности также уже при  $p = 10$  практически не превышают значения 0,01.

Таблица 2. Результаты восстановления отсчетов: прямое прогнозирование

Частота дискретизации $F_s = 100f_s = 100$								
$m$	$k+m$	$x_{k+m}$	Порядок модели					
			$p = 5$		$p = 10$		$p = 15$	
			$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$
1	1201	0,4513	0,4508	-0,0011	0,4511	-0,0004	0,4511	-0,0003
2	1202	0,4681	0,4670	-0,0023	0,4677	-0,0009	0,4678	-0,0007
3	1203	0,4848	0,4828	-0,0041	0,4841	-0,0014	0,4842	-0,0011
4	1204	0,5011	0,4981	-0,0060	0,5001	-0,0020	0,5003	-0,0016
5	1205	0,5172	0,5130	-0,0081	0,5158	-0,0027	0,5161	-0,0021
6	1206	0,5330	0,5273	-0,0107	0,5311	-0,0036	0,5315	-0,0028
7	1207	0,5485	0,5412	-0,0133	0,5461	-0,0044	0,5466	-0,0035
9	1208	0,5636	0,5545	-0,0161	0,5606	-0,0053	0,5612	-0,0043
9	1209	0,5784	0,5673	-0,0192	0,5747	-0,0064	0,5753	-0,0052
10	1210	0,5927	0,5795	-0,0223	0,5883	-0,0074	0,5890	-0,0062
11	1211	0,6066	0,5911	-0,0256	0,6014	-0,0086	0,6022	-0,0073
12	1212	0,6201	0,6020	-0,0292	0,6140	-0,0098	0,6149	-0,0085
13	1213	0,6332	0,6123	-0,0330	0,6260	-0,0114	0,6270	-0,0098
14	1214	0,6458	0,6220	-0,0369	0,6374	-0,0130	0,6385	-0,0112
15	1215	0,6578	0,6310	-0,0407	0,6483	-0,0144	0,6494	-0,0128
$\delta_{\text{MSE}}^x$			0,01363		0,00470		0,00405	
$\delta_{\text{MAE}}^x$			0,01082		0,00370		0,00314	

Таблица 3. Результаты восстановления отсчетов: обратное прогнозирование

Частота дискретизации $F_s = 100f_s = 100$								
$m$	$k+m$	$x_{k+m}$	Порядок модели					
			$p = 5$		$p = 10$		$p = 15$	
			$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$
1	1201	0,4513	0,4508	-0,0011	0,4511	-0,0005	0,4511	-0,0005
2	1202	0,4681	0,4669	-0,0026	0,4677	-0,0007	0,4678	-0,0006
3	1203	0,4848	0,4826	-0,0045	0,4841	-0,0015	0,4842	-0,0012
4	1204	0,5011	0,4979	-0,0064	0,5001	-0,0018	0,5003	-0,0016
5	1205	0,5172	0,5127	-0,0087	0,5159	-0,0027	0,5161	-0,0021
6	1206	0,5330	0,5270	-0,0113	0,5312	-0,0034	0,5315	-0,0028
7	1207	0,5485	0,5408	-0,0140	0,5462	-0,0042	0,5465	-0,0036
8	1208	0,5636	0,5540	-0,0170	0,5608	-0,0052	0,5611	-0,0044
9	1209	0,5784	0,5667	-0,0202	0,5749	-0,0060	0,5753	-0,0054
10	1210	0,5927	0,5787	-0,0236	0,5885	-0,0072	0,5889	-0,0064

Частота дискретизации $F_s = 100f_s = 100$								
$m$	$k+m$	$x_{k+m}$	Порядок модели					
			$p = 5$		$p = 10$		$p = 15$	
			$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$
11	1211	0,6066	0,5902	-0,0270	0,6016	-0,0083	0,6021	-0,0074
12	1212	0,6201	0,6010	-0,0308	0,6142	-0,0094	0,6147	-0,0087
13	1213	0,6332	0,6112	-0,0347	0,6263	-0,0110	0,6268	-0,0101
14	1214	0,6458	0,6207	-0,0389	0,6377	-0,0126	0,6382	-0,0118
15	1215	0,6578	0,6296	-0,0429	0,6486	-0,0141	0,6491	-0,0132
		$\delta_{MSE}^x$	0,01438		0,00453		0,00418	
		$\delta_{MAE}^x$	0,01143		0,00355		0,00323	

Таблица 4. Результаты восстановления отсчетов: прямое и обратное прогнозирование

Частота дискретизации $F_s = 100f_s = 100$								
$m$	$k+m$	$x_{k+m}$	Порядок модели					
			$p = 5$		$p = 10$		$p = 15$	
			$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$
1	1201	0,4513	0,4509	-0,0008	0,4512	-0,0001	0,4512	-0,0001
2	1202	0,4681	0,4671	-0,0021	0,4678	-0,0004	0,4679	-0,0003
8	1203	0,4848	0,4828	-0,0042	0,4842	-0,0013	0,4844	-0,0011
4	1204	0,5011	0,4981	-0,0059	0,5003	-0,0015	0,5005	-0,0012
5	1205	0,5172	0,5130	-0,0081	0,5161	-0,0021	0,5163	-0,0016
6	1206	0,5330	0,5273	-0,0105	0,5315	-0,0027	0,5318	-0,0022
7	1207	0,5485	0,5412	-0,0135	0,5465	-0,0037	0,5468	-0,0031
3	1208	0,5636	0,5544	-0,0165	0,5611	-0,0046	0,5615	-0,0039
9	1209	0,5784	0,5671	-0,0193	0,5753	-0,0053	0,5758	-0,0044
10	1210	0,5927	0,5793	-0,0228	0,5890	-0,0064	0,5895	-0,0055
11	1211	0,6066	0,5908	-0,0260	0,6021	-0,0073	0,6028	-0,0063
12	1212	0,6201	0,6017	-0,0299	0,6148	-0,0087	0,6155	-0,0076
13	1213	0,6332	0,6119	-0,0337	0,6269	-0,0100	0,6277	-0,0087
14	1214	0,6458	0,6215	-0,0375	0,6385	-0,0112	0,6394	-0,0098
15	1215	0,6578	0,6304	-0,0416	0,6494	-0,0127	0,6504	-0,0112
		$\delta_{MSE}^x$	0,01388		0,00408		0,00355	
		$\delta_{MAE}^x$	0,01098		0,00317		0,00271	

В процессе тестовых испытаний, связанных с исследованием способности алгоритмов прогнозировать и восстанавливать значения отсчетов, когда частоты гармонических компонент сигнала были бы близки к частоте дискретизации, формирование последовательности осуществлялось с предельно низкой частотой. Для гармонического состава модели сигнала, приведенного в таблице 1, согласно теореме Котельникова (Найквиста) она была равна  $F_s = 2f_s = 2$ , т.е. интервал дискретизации был равен  $\Delta t = 0,5$ . В частности, отсчеты последовательности в количестве  $M = 15$  с индексами  $n \in [121; 135]$  рассматривались в качестве проблемного участка. Здесь  $k = 120$ . На рисунке 5 приведен фрагмент последовательности модели сигнала с учетом места нахождения этого участка, а на рисунке 6 приведен фрагмент оценки КФ, вычисленной по отсчетам данной последовательности.

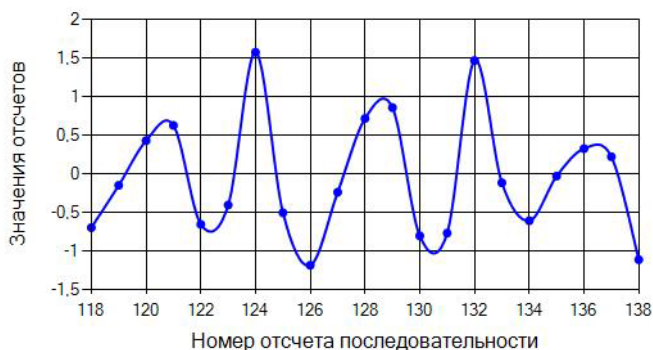


Рис. 5. Проблемный участок последовательности;  $F_s = 2f_s = 2$

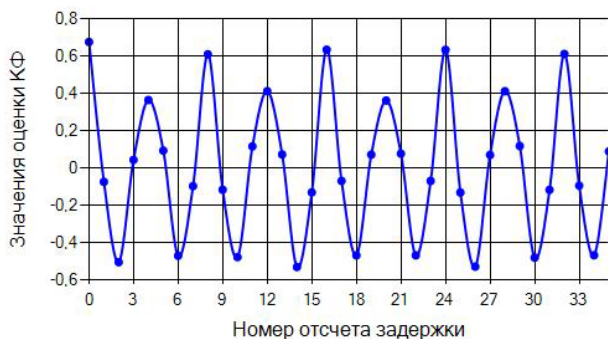


Рис. 6. Фрагмент оценки КФ;  $F_s = 2f_s = 2$



В таблице 5 представлены результаты тестовых экспериментов по восстановлению проблемного участка последовательности для  $n \in [121; 135]$  при прямом, обратном и совместном прогнозировании для случая, когда порядок модели восстановления  $p = 10$ . В пределах всего проблемного участка значения относительной погрешности для любой процедуры восстановления находятся на достаточно низком уровне. Среднеквадратическая и средняя абсолютная погрешности практически находятся в пределах значения 0,01.

Таблица 5. Результаты восстановления отсчетов: порядок модели  $p = 10$ 

Частота дискретизации $F_s = 2f_5 = 2$								
$m$	$k+m$	$x_{k+m}$	Вид прогнозирования					
			Прямое		Обратное		Прямое и обратное	
			$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$	$\hat{x}_{k+m}$	$\delta_{k+m}^x$
1	121	0,6282	0,6393	0,0176	0,6394	0,0180	0,6393	0,0176
2	122	-0,6538	-0,6296	-0,0370	-0,6296	-0,0370	-0,6296	- 0,0369
3	123	-0,4033	-0,4232	0,0492	-0,4233	0,0492	-0,4232	0,0493
4	124	1,5754	1,5712	-0,0027	1,5712	-0,0027	1,5711	- 0,0027
5	125	-0,5028	-0,5212	0,0367	-0,5214	0,0368	-0,5212	0,0366
6	126	-1,1841	-1,1788	-0,0044	-1,1789	-0,0044	-1,1788	- 0,0045
7	127	-0,2373	-0,2284	-0,0379	-0,2283	-0,0374	-0,2283	- 0,0378
8	128	0,7175	0,7128	-0,0065	0,7130	-0,0060	0,7128	- 0,0064
9	129	0,8594	0,8555	-0,0045	0,8555	-0,0043	0,8555	- 0,0046
10	130	-0,8030	-0,7886	-0,0179	-0,7885	-0,0180	-0,7885	- 0,0181
11	131	-0,7678	-0,7792	0,0148	-0,7792	0,0151	-0,7791	0,0148
12	132	1,4689	1,4869	0,0123	1,4872	0,0125	1,4869	0,0123
13	133	-0,1153	-0,1167	0,0121	-0,1168	0,0130	-0,1168	0,0127
14	134	-0,6073	-0,6051	-0,0036	-0,6052	-0,0038	-0,6051	- 0,0036
15	135	-0,0292	-0,0318	0,0885	-0,0319	0,0989	-0,0318	0,0865
$\delta_{MSE}^x$			0,01233		0,012402		0,01234	
$\delta_{MAE}^x$			0,01004		0,010087		0,01006	

Тестовые эксперименты проводились также с учетом увеличения длительности проблемного участка, в пределах которого необходимо осуществить процедуру восстановления отсчетов. Целью этих экспериментов была оценка точности восстановления значений утраченных отсчетов по мере увеличения временного интервала проблемного участка при неизменном значении порядка модели. В данном случае также рассматривалась последовательность, полученная в результате дискретизации с предельно низкой частотой  $F_s = 2f_s = 2$  модели сигнала с параметрами, приведенными в таблице 1. В частности, в процессе имитации увеличения длительности проблемного участка использовались отсчеты с индексами  $n \in [121; 220]$ . В ходе проведения экспериментов было принято, что  $k = 120$  и остается постоянным, а участок с проблемными отсчетами последовательно увеличивался с начального значения  $M = 10$  до  $M = 100$ . При этом интервал увеличения длительности проблемного участка был равен десяти отсчетам. К примеру, в таблице 6 для порядка модели  $p = 10$  приведены значения среднеквадратической и средней абсолютной погрешностей, характеризующие точность вычисления оценок значений отсчетов при прямом прогнозировании в зависимости от увеличения длительности проблемного участка. Эти погрешности практически незначительно превышают значение 0,01.

Таблица 6. Среднеквадратическая и средняя абсолютная погрешности восстановления значений отсчетов при прямом прогнозировании в зависимости от увеличения длительности проблемного участка

Частота дискретизации $F_s = 2f_s = 2$ . Порядок модели $p = 10$ .						
M	Вид прогнозирования					
	Прямое		Обратное		Прямое и обратное	
	$\delta_{MSE}^x$	$\delta_{MAE}^x$	$\delta_{MSE}^x$	$\delta_{MAE}^x$	$\delta_{MSE}^x$	$\delta_{MAE}^x$
10	0,01231	0,01008	0,01230	0,01007	0,01221	0,01002
20	0,01242	0,01023	0,01243	0,01023	0,01235	0,01017
30	0,01251	0,01032	0,01255	0,01032	0,01245	0,01026
40	0,01231	0,01008	0,01230	0,01008	0,01221	0,01000
50	0,01255	0,01036	0,01254	0,01036	0,01246	0,01029
60	0,01262	0,01031	0,01261	0,01030	0,01250	0,01023
70	0,01272	0,01049	0,01271	0,01049	0,01262	0,01041
80	0,01347	0,01150	0,01345	0,01149	0,01332	0,01131
90	0,01352	0,01132	0,01350	0,01131	0,01337	0,01118
100	0,01302	0,01062	0,01301	0,01062	0,01289	0,01051

Полученные результаты тестовых экспериментов, позволяют сделать вывод о том, что с метрологической точки зрения разработанное алгоритмическое обеспечение на основе построения модели скользящего среднего и оценки корреляционной связи между отсчетами при прямом и обратном прогнозировании позволяет осуществлять восстановление значений отсчетов дискретной последовательности сигнала с достаточно низкой погрешностью.

В реальных условиях среда распространения сигнала может быть подвержена воздействию внешних фоновых шумов. Это может привести к эффекту наложения шума на полезный сигнал и оказать негативное влияние на его форму. В этом случае можно считать, что восстановленные значения содержат оценку и шумовой составляющей в отсчетах сигнала. Вследствие этого в процессе последующей цифровой обработки восстановленной дискретной последовательности по возможности следует учитывать степень коррелированности полезного сигнала и шумов среды его распространения. При этом необходимо иметь в виду, что КФ случайного шума по мере увеличения временного сдвига затухает и ее можно считать отличной от нуля только в пределах интервала корреляции его значений.

**5. Заключение.** В статье рассмотрена задача восстановления численных значений отсчетов на проблемных участках дискретной последовательности, полученной в результате равномерной дискретизации во времени непрерывного сигнала. Задача решалась исходя из того, что исходный сигнал можно рассматривать как стационарный или стационарный хотя бы в широком смысле (квазистационарный) в пределах интервала времени его анализа. Разработка математического и соответствующего ему алгоритмического обеспечения для восстановления значений отсчетов сигнала осуществлена на основе построения модели скользящего среднего и оценки их корреляционной связи во времени при прямом, обратном и совместном прогнозировании. Необходимая для восстановления последовательность оценок КФ сигнала вычисляется по отсчетам последовательности с известными значениями. С учетом выполнения условия стационарности сигнала это можно сделать на любом участке последовательности независимо от места нахождения восстанавливаемого участка. При этом полученные оценки отсчетов КФ могут использоваться как для прямого, так и для обратного прогнозирования. Более того, даже если проблемных участков будет несколько, достаточно только один раз вычислить необходимую для их восстановления последовательность оценок КФ. Единственным условием является только то, что число отсчетов участка

последовательности с известными значениями, на котором будут вычисляться отсчеты КФ, должно быть достаточным для этого.

Значения весовых коэффициентов модели скользящего среднего при прямом, обратном и совместном прогнозировании находятся из критерия минимума среднеквадратической погрешности, среднее значение которой используется в качестве критерия выбора ее порядка.

Полученные математические соотношения стали основой для разработки вычислительных алгоритмов, определяющих процедуры восстановления утраченных значений отсчетов. Данные алгоритмы являются неитеративными. На их основе с использованием компонентно-ориентированного программирования разработано программное обеспечение, представляющее собой функционально завершенный модуль. Данный модуль предназначен для работы в составе метрологически значимого программного обеспечения многофункциональных систем цифровой обработки сигналов.

Тестовые испытания и функциональные проверки разработанного алгоритмического обеспечения и программного модуля проводились на основе имитационного моделирования. Полученные результаты показали, что вычисление оценок значений утраченных отсчетов на проблемных участках последовательности сигнала осуществляется с достаточно низкой погрешностью, как при прямом, так и при обратном прогнозировании, а также при их совместном использовании.

На практике выбор конкретного алгоритма восстановления последовательности на основе прямого или обратного прогнозирования будет определяться исходя из реальных условий, в которых приходится осуществлять восстанавливающие процедуры. В частности, если предыдущих отсчетов с известными значениями недостаточно для осуществления прямого прогнозирования, то реализуется процедура обратного прогнозирования. Может возникнуть ситуация, когда восстановлению подлежат несколько проблемных участков последовательности, находящихся близко по отношению друг к другу. Может оказаться так, что для некоторых из этих участков будет невозможно осуществить процедуру восстановления утраченных значений отсчетов непосредственно ни по предыдущим и ни по последующим отсчетам с известными значениями. В принципе в такой ситуации можно использовать оба вида прогнозирования. Однако участки отсчетов с известными значениями могут быть достаточно удалены от проблемных участков. В этом случае более целесообразно

использовать алгоритм восстановления значений отсчетов одновременно на основе прямого и обратного прогнозирования.

### Литература

1. Zhang T., Ren J., Li J., Nguyen L.H., Stoica P. Joint RFI Mitigation and Radar Echo Recovery for One-Bit UWB Radar // *Signal Processing*. 2022. vol. 193. no. 108409.
2. Mardani D., Atia G.K., Abouraddy A.F. Signal Reconstruction from Interferometric Measurements under Sensing Constraints // *Signal Processing*. 2019. vol. 155. pp. 323–333.
3. Zhang Y., Yong G. Source Number of Single-Channel Signals Intelligent Estimation via Signal Reconstruction // *Digital Signal Processing*. 2023. vol. 140(16). no. 104100.
4. Семченков С.М., Жбанов И.Л., Абраменков А.В., Коваленков А.Н., Макаров М.С., Печенев Е.А. Методы обеспечения устойчивости восстановления сигнала в радиолокационных измерениях // *Электромагнитные волны и электронные системы*. 2020. Т. 25. № 5. С. 50–66.
5. Amini F., Hedayati Y., Zanddizari H. Exploiting the Inter-Correlation of Structural Vibration Signals for Data Loss Recovery: A Distributed Compressive Sensing Based Approach // *Mechanical Systems and Signal Processing*. 2021. vol. 152. no. 107473.
6. Thadikemalla V.S.G., Gandhi A.S. A Data Loss Recovery Technique Using Compressive Sensing for Structural Health Monitoring Applications // *KSCSE Journal of Civil Engineering*. 2018. vol. 22. no. 12. pp. 5084–5093.
7. Li Y., Tse P.W., Wang X. Recovery of Vibration Signal Based on a Super-Exponential Algorithm // *Journal of Sound and Vibration*. 2008. vol. 311. no. 1-2. pp. 537–553.
8. Liu F., Zhang A., Du R., Xu J., Hu Z. HTR-CTO Algorithm for Wireless Data Recovery // *Information Sciences*. 2022. vol. 587. pp. 515–534.
9. Yang L., Wang H., Qian H. An ADMM-ResNet for Data Recovery in Wireless Sensor Networks with Guaranteed Convergence // *Digital Signal Processing*. 2021. vol. 111. no. 102956.
10. Wu H., Xian J., Wang J., Khandge S., Mohapatra P. Missing Data Recovery Using Reconstruction in Ocean Wireless Sensor Networks // *Computer Communications*. 2018. vol. 132. pp. 1–9.
11. Поршнев С.В., Кусайкин Д.В. Восстановление неравномерно дискретизированных сигналов с неизвестными значениями координат узлов временной сетки // *Успехи современной радиоэлектроники*. 2015. № 6. С. 3–35.
12. Khan N.A., Ali S. Robust Sparse Reconstruction of Signals with Gapped Missing Samples from Multi-Sensor Recordings // *Digital Signal Processing*. 2022. vol. 123. no. 103392.
13. Aceska R., Bouchot J.-L., Li S. Local Sparsity and Recovery of Fusion Frame Structured Signals // *Signal Processing*. 2020. vol. 174. no. 107615.
14. Stankovic L., Stankovic S., Amin M. Missing samples analysis in signals for applications to L-estimation and compressive sensing // *Signal Processing*. 2014. vol. 94. pp. 401–408.
15. Aldroubi A., Leonetti C. Non-Uniform Sampling and Reconstruction from Sampling Sets with Unknown Jitter // *Sampling Theory in Signal and Image Processing*. 2008. vol. 7. no. 2. pp. 187–195.
16. Nordio A., Chiasserini C-F., Viterbo E. Signal Reconstruction Errors in Jittered Sampling // *IEEE Transactions on signal Processing*. 2009. vol. 57. no. 12. pp. 4711–4718.

17. Maymon S., Oppenheim A.V. Sinc Interpolation of Nonuniform Samples // IEEE Transactions on Signal Processing. 2011. vol. 59. no. 10. pp. 4745–4758.
18. Andras I., Dolinsky P., Michaeli L., Saliga J. A Time Domain Reconstruction Method of Randomly Sampled Frequency Sparse Signal // Measurement. 2018. vol. 127. pp. 68–77.
19. Cui A, Zhang L., He H., Wen M. A Truncated Approximate Difference Algorithm for Sparse Signal Recovery // Digital Signal Processing. 2023. vol. 141. no. 104191.
20. Bilinskis I. Digital Alias-free Signal Processing // Wiley. 2007. 454 p.
21. Eamaz A., Yeganegi F., Soltanalian M. Covariance Recovery for One-Bit Sampled Stationary Signals with Time-Varying Sampling Thresholds // Signal Processing. 2023. vol. 206. no. 108899.
22. Якимов В.Н., Машков А.В. Знаковый алгоритм анализа спектра амплитуд и восстановления гармонических составляющих сигналов в условиях присутствия некоррелированных фоновых шумов // Научное приборостроение. 2017. Т. 27. № 2. С. 83–90.
23. Wanga P., Yang H., Yea Z. 1-Bit direction of arrival estimation via improved complex-valued binary iterative hard thresholding // Digital Signal Processing. 2022. vol. 120. no. 103265.
24. Guoa L.-B., Donga C.-X., Donga Y.-Y., Sunb T., Maoa Y. One-bit LFM signal recovery: A consistency algorithm with one-sided weighted quadratic penalty // Digital Signal Processing. 2022. vol. 127. no. 103575.
25. Якимов В.Н., Машков А.В. Цифровое оценивание моментов корреляционной функции на основе знакового аналого-стохастического квантования случайного процесса // Измерительная техника. 2016. № 1. С. 11–13.
26. Lu Y.M., Vetterli M. Multichannel Sampling with Unknown Gains and Offsets: A Fast Reconstruction Algorithm // Proc. of Allerton Conference on Communication, Control and Computing. 2010. 7 p.
27. Choe C.-G., Pak J.-H., Rim C.-S. Joint. Near-Isometry and Optimal Sparse Recovery: Nonuniform Recovery from Multi-Sensor Measurements // Signal Processing. 2023. vol. 208. no. 108980.
28. Liu N., Tao R., Wang R., Deng Y., Li N., Zhao S. Signal Reconstruction from Recurrent Samples in Fractional Fourier Domain and Its application in Multichannel SAR // Signal Processing. 2017. vol. 131. pp. 288–299.
29. Wang H., Yang S., Liu Y., Li Q. Compressive Sensing Reconstruction for Rolling Bearing Vibration Signal Based on Improved Iterative Soft Thresholding Algorithm // Measurement. 2023. vol. 210. no. 112528.
30. Sejdic E., Orovic I., Stankovic S. Compressive sensing meets time-frequency: An overview of recent advances in time-frequency processing of sparse signals // Digital Signal Processing. 2018. vol. 77. pp. 22–35.
31. Tang Q., Jiang Y., Xin J., Liao G., Zhou J. Yang X. A Novel Method for the Recovery of Continuous Missing Data Using Multivariate Variational Mode Decomposition and Fully Convolutional Networks // Measurement. 2023. vol. 220. no. 113366.
32. Жукова Н.А., Соколов И.С. Метод восстановления структуры группового телеметрического сигнала на основе графовой модели // Труды СПИИРАН. 2010. vol. 2(13). С. 45–66.
33. Khan N.A., Ali S. Reconstruction of gapped missing samples based on instantaneous frequency and instantaneous amplitude estimation // Signal Processing. 2022. vol. 193. no. 108429.
34. Dokuchaev N. On Recovery of Discrete Time Signals from Their Periodic Subsequences // Signal Processing. 2019. vol. 162. pp. 180–188.

35. Annaby M.H., Al-Abdi I.A., Abou-Dina M.S., Ghaleb A.F. Regularized Sampling Reconstruction of Signals in the Linear Canonical Transform Domain // *Signal Processing*. 2022. vol. 198. no. 108569.
36. Yue C., Liang J., Qu B., Han Y., Zhu Y., Crisalle O.D. A Novel Multiobjective Optimization Algorithm for Sparse Signal Reconstruction // *Signal Processing*. 2020. vol. 167. no. 107292.
37. Wijenayake C., Scutts J., Ignjatovic A. Signal recovery algorithm for 2-level amplitude sampling using chromatic signal approximations // *Signal Processing*. 2018. vol. 153. pp. 143–152.
38. Катковник В.Я. Непараметрическая идентификация и сглаживание данных: метод локальной аппроксимации // М.: Главная редакция физико-математической литературы. 1985. 336 с.
39. Якимов В.Н. Восстановление дискретной временной последовательности сигнала на основе локальной аппроксимации с использованием ряда Фурье по ортогональной системе тригонометрических функций // *Информатика и автоматизация*. 2022. Т. 21. № 5. С. 1016–1043.
40. Marple S.L.Jr. *Digital Spectral Analysis: Second Edition* // Mineola, New York: Dover Publications, Inc. 2019. 409 p.
41. Proakis J.G., Manolakis D.G. *Digital Signal Processing* // Pearson Education Limited, 2014. 1014 p.
42. ГОСТ Р 8.883-2015 Государственная система обеспечения единства измерений. Программное обеспечение средств измерений. Алгоритмы обработки, хранения, защиты и передачи измерительной информации. Методы испытаний. Введ. 2016-03-01. М.: Стандартинформ, 2018. 19 с.
43. ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию. Введ. 2003-07-01. М.: Госстандарт России, 2005. 63 с.
44. ГОСТ 8.654-2015 Государственная система обеспечения единства измерений. Требования к программному обеспечению средств измерений. Основные положения. Введ. 2016-03-01. М.: Стандартинформ, 2015. 11 с.
45. Yakimov V.N., Gorbachev O.V. Firmware of the Amplitude Spectrum Evaluating System for Multicomponent Processes // *Instruments and Experimental Techniques*. 2013. vol. 56. no. 5. pp. 540–545.
46. Yakimov V.N., Zaberzhinskij B.E., Mashkov A.V., Bukanova Yu.V. Multi-threaded Approach to Software High-speed Algorithms for Spectral Analysis of Multi-component Signals // *XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP)*. 2019. IEEE. pp. 698–701.
47. Якимов В.Н., Машков А.В., Желонкин А.В. Специализированное программное обеспечение измерительной системы для оперативного оценивания спектрального состава многокомпонентных процессов // *Программные продукты и системы*. 2019. Т. 32. № 1. С. 159–166.
48. ГОСТ Р 57700.22-2020 Компьютерные модели и моделирование. Классификация. Введ. 2021-06-01. М.: Стандартинформ, 2020. 7 с.

**Якимов Владимир Николаевич** — д-р техн. наук, профессор кафедры, кафедра автоматизации и управления технологическими процессами института автоматики и информационных технологий, Самарский государственный технический университет (СамГТУ). Область научных интересов: методы и средства статистических измерений, диагностика технического состояния и испытания систем, цифровая обработка сигналов, корреляционный и спектральный анализ сигналов, математическое и имитационное моделирование. Число научных публикаций — 300. [yvnr@hotmail.com](mailto:yvnr@hotmail.com); улица Молодогвардейская, 244, 443100, Самара, Россия; р.т.: +7(846)279-0354.

V. YAKIMOV

**RECOVERY OF DISCRETE-TIME SIGNAL BASED ON THE MOVING AVERAGE MODEL AND ESTIMATION OF THE SAMPLES CORRELATION IN FORWARD AND REVERSE FORECASTING**

*Yakimov V. Recovery of Discrete-Time Signal Based on the Moving Average Model and Estimation of the Samples Correlation in Forward and Reverse Forecasting.*

**Abstract.** The article discusses the development of mathematical support for the recovery of the values of discrete-time sequence samples obtained as a result of uniform sampling of a continuous signal. The recovery problem of discrete-time sequence samples is solved for a signal that can be considered stationary or stationary at least in a broad sense (quasi-stationary). The development of mathematical support for the recovery of the values of signal samples was carried out on the basis of constructing a moving average model and estimating the correlation of signal samples over time with forward and reverse forecasting. Estimates of the signal correlation function necessary to recover sample sections with lost values are calculated from samples with known values. Correlation function estimates can be calculated regardless of the location of the recovery area when the condition of stationarity of the signal is met. The obtained estimates of the correlation function samples can be used for both forward and reverse forecasting. Moreover, even if it is necessary to recover several problem sections, it is enough to calculate only once the sample of correlation function estimates necessary for their restoration. The resulting mathematical solution to the problem became the basis for the development of algorithmic support. Test tests and functional checks of the algorithmic support were carried out on the basis of simulation using a signal model representing an additive sum of harmonic components with random initial phases. The simulation results showed that the calculation of estimates of the lost sample values is carried out with a fairly low error, both in forward and reverse forecasting, as well as when they are used together. In practice, the choice of a sequence recovery algorithm based on forward or reverse forecasting will be determined based on the actual conditions of its processing. In particular, if previous samples with known values are not enough to carry out forward forecasting, then the reverse forecasting procedure is implemented and vice versa. The developed algorithmic support can be implemented in the form of metrologically significant software for digital signal processing systems.

**Keywords:** discrete-time signals, sampled sequence, signal recovery, forward and reverse forecasting, correlation relationship.

**References**

1. Zhang T., Ren J., Li J., Nguyen L.H., Stoica P. Joint RFI Mitigation and Radar Echo Recovery for One-Bit UWB Radar. *Signal Processing*. 2022. vol. 193. no. 108409.
2. Mardani D., Atia G.K., Abouraddy A.F. Signal Reconstruction from Interferometric Measurements under Sensing Constraints. *Signal Processing*. 2019. vol. 155. pp. 323–333.
3. Zhang Y., Yong G. Source Number of Single-Channel Signals Intelligent Estimation via Signal Reconstruction. *Digital Signal Processing*. 2023. vol. 140(16). no. 104100.
4. Semchenkov S.M., ZHbanov I.L., Abramnikov A.V., Kovalenkov A.N., Makarov M.S., Pechevnev E.A. [Methods for ensuring the stability of signal recovery in radar measurements.]. *Elektromagnitnye volny i elektronnyye sistemy – Electromagnetic waves and electronic systems*. 2020. vol. 25. no. 5. pp. 50–66.



5. Amini F., Hedayati Y., Zanddzari H. Exploiting the Inter-Correlation of Structural Vibration Signals for Data Loss Recovery: A Distributed Compressive Sensing Based Approach. *Mechanical Systems and Signal Processing*. 2021. vol. 152. no. 107473.
6. Thadikemalla V.S.G., Gandhi A.S. A Data Loss Recovery Technique Using Compressive Sensing for Structural Health Monitoring Applications. *KSCE Journal of Civil Engineering*. 2018. vol. 22. no. 12. pp. 5084–5093.
7. Li Y., Tse P.W., Wang X. Recovery of Vibration Signal Based on a Super-Exponential Algorithm. *Journal of Sound and Vibration*. 2008. vol. 311. no. 1-2. pp. 537–553.
8. Liu F., Zhang A., Du R., Xu J., Hu Z. HTR-CTO Algorithm for Wireless Data Recovery. *Information Sciences*. 2022. vol. 587. pp. 515–534.
9. Yang L., Wang H., Qian H. An ADMM-ResNet for Data Recovery in Wireless Sensor Networks with Guaranteed Convergence. *Digital Signal Processing*. 2021. vol. 111. no. 102956.
10. Wu H., Xian J., Wang J., Khandge S., Mohapatra P. Missing Data Recovery Using Reconstruction in Ocean Wireless Sensor Networks. *Computer Communications*. 2018. vol. 132. pp. 1–9.
11. Porshnev S.V., Kusaykin D.V. [Reconstruction of non-uniform sampled discrete-time signals with unknown sampling locations]. *Uspekhi sovremennoy radioelektroniki – Journal Achievements of Modern Radioelectronics*. 2015. no. 6. pp. 3–35. (In Russ.).
12. Khan N.A., Ali S. Robust Sparse Reconstruction of Signals with Gapped Missing Samples from Multi-Sensor Recordings. *Digital Signal Processing*. 2022. vol. 123. no. 103392.
13. Aceska R., Bouchot J.-L., Li S. Local Sparsity and Recovery of Fusion Frame Structured Signals. *Signal Processing*. 2020. vol. 174. no. 107615.
14. Stankovic L., Stankovic S., Amin M. Missing samples analysis in signals for applications to L-estimation and compressive sensing. *Signal Processing*. 2014. vol. 94. pp. 401–408.
15. Aldroubi A., Leonetti C. Non-Uniform Sampling and Reconstruction from Sampling Sets with Unknown Jitter. *Sampling Theory in Signal and Image Processing*. 2008. vol. 7. no. 2. pp. 187–195.
16. Nordio A., Chiasserini C-F., Viterbo E. Signal Reconstruction Errors in Jittered Sampling. *IEEE Transactions on signal Processing*. 2009. vol. 57. no. 12. pp. 4711–4718.
17. Maymon S., Oppenheim A.V. Sinc Interpolation of Nonuniform Samples. *IEEE Transactions on Signal Processing*. 2011. vol. 59. no. 10. pp. 4745–4758.
18. Andras I., Dolinsky P., Michaeli L., Saliga J. A Time Domain Reconstruction Method of Randomly Sampled Frequency Sparse Signal. *Measurement*. 2018. vol. 127. pp. 68–77.
19. Cui A, Zhang L., He H., Wen M. A Truncated Approximate Difference Algorithm for Sparse Signal Recovery. *Digital Signal Processing*. 2023. vol. 141. no. 104191.
20. Bilinskis I. *Digital Alias-free Signal Processing*. Wiley, 2007. 454 p.
21. Eamaz A., Yeganegi F., Soltanalian M. Covariance Recovery for One-Bit Sampled Stationary Signals with Time-Varying Sampling Thresholds. *Signal Processing*. 2023. vol. 206. no. 108899.
22. Yakimov V.N., Mashkov A.V. [The binary algorithm for the analysis of the spectrum amplitude and recover of harmonic components signals in the presence of uncorrelated background noise]. *Nauchnoe priborostroenie – Scientific Instrument Making*. 2017. vol. 27. no.2. pp. 83–90. (In Russ.).
23. Wanga P., Yanga H., Yea Z. 1-Bit direction of arrival estimation via improved complex-valued binary iterative hard thresholding. *Digital Signal Processing*. 2022. vol. 120. no. 103265.

24. Guoa L.-B., Donga C.-X., Donga Y.-Y., Sunb T., Maoa Y. One-bit LFM signal recovery: A consistency algorithm with one-sided weighted quadratic penalty. *Digital Signal Processing*. 2022. vol. 127. no. 103575.
25. Yakimov V.N., Mashkov A.V. Digital Estimation of Correlation Function Moments Using Analog-Stochastic Sign Quantization of a Random Process. *Measurement Techniques*. 2016. vol. 59. no. 1. pp. 12–15.
26. Lu Y.M., Vetterli M. Multichannel Sampling with Unknown Gains and Offsets: A Fast Reconstruction Algorithm. *Proc. of Allerton Conference on Communication, Control and Computing*. 2010. 7 p.
27. Choe C.-G., Pak J.-H., Rim C.-S. Joint. Near-Isometry and Optimal Sparse Recovery: Nonuniform Recovery from Multi-Sensor Measurements. *Signal Processing*. 2023. vol. 208. no. 108980.
28. Liu N., Tao R., Wang R., Deng Y., Li N., Zhao S. Signal Reconstruction from Recurrent Samples in Fractional Fourier Domain and Its application in Multichannel SAR. *Signal Processing*. 2017. vol. 131. pp. 288–299.
29. Wang H., Yang S., Liu Y., Li Q. Compressive Sensing Reconstruction for Rolling Bearing Vibration Signal Based on Improved Iterative Soft Thresholding Algorithm. *Measurement*. 2023. vol. 210. no. 112528.
30. Sejdic E., Orovic I., Stankovic S. Compressive sensing meets time-frequency: An overview of recent advances in time-frequency processing of sparse signals. *Digital Signal Processing*. 2018. vol. 77. pp. 22–35.
31. Tang Q., Jiang Y., Xin J., Liao G., Zhou J., Yang X. A Novel Method for the Recovery of Continuous Missing Data Using Multivariate Variational Mode Decomposition and Fully Convolutional Networks. *Measurement*. 2023. vol. 220. no. 113366.
32. Zhukova N.A., Sokolov I.S. [Method of reconstructing the structure of the group telemetric signals based on the graph model]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2010. no. 13. pp. 45–66. (In Russ.).
33. Khan N.A., Ali S. Reconstruction of gapped missing samples based on instantaneous frequency and instantaneous amplitude estimation. *Signal Processing*. 2022. vol. 193. no. 108429.
34. Dokuchaev N. On Recovery of Discrete Time Signals from Their Periodic Subsequences. *Signal Processing*. 2019. vol. 162. pp. 180–188.
35. Annaby M.H., Al-Abdi I.A., Abou-Dina M.S., Ghaleb A.F. Regularized Sampling Reconstruction of Signals in the Linear Canonical Transform Domain. *Signal Processing*. 2022. vol. 198. no. 108569.
36. Yue C., Liang J., Qu B., Han Y., Zhu Y., Crisalle O.D. A Novel Multiobjective Optimization Algorithm for Sparse Signal Reconstruction. *Signal Processing*. 2020. vol. 167. no. 107292.
37. Wijenayake C., Scutts J., Ignjatovic A. Signal recovery algorithm for 2-level amplitude sampling using chromatic signal approximations. *Signal Processing*. 2018. vol. 153. pp. 143–152.
38. Katkovnik V.Ya. Neparаметricheskaya identifikatsiya i sglazhivaniye dannykh: metod lokal'noy approksimatsii [Nonparametric identification and data smoothing: local approximation method]. Moscow: Glavnaya redaktsiya fiziko-matematicheskoy literatury. 1985. 336 p. (In Russ.).
39. Yakimov V.N. [Discrete Time Sequence Reconstruction of a Signal Based on Local Approximation Using a Fourier Series by an Orthogonal System of Trigonometric Functions]. *Informatika i Avtomatizatsiya – Informatics and Automation*. 2022. vol. 21. no. 5. pp. 1016–1043.
40. Marple S.L.Jr. *Digital Spectral Analysis: Second Edition*. Mineola, New York: Dover Publications, Inc. 2019. 409 p.

41. Proakis J.G., Manolakis D.G. Digital Signal Processing. Pearson Education Limited, 2014. 1014 p.
42. State Standard RV 8.883-2015. Gosudarstvennaya sistema obespecheniya yedinstva izmereniy. Programmnoye obespecheniye sredstv izmereniy. Algoritmy obrabotki, khraneniya, zashchity i peredachi izmeritel'noy informatsii [State system for ensuring the uniformity of measurements. Software of measuring instruments. Algorithms of processing, storage, protection and transmission of measuring information. Test methods]. Moscow: Standartinform Publ., 2018. (In Russ.).
43. State Standard RV 51904-2002. Programmnoye obespecheniye vstroyennykh sistem. Obshchiye trebovaniya k razrabotke i dokumentirovaniyu [Embedded system software. General requirements for development and documentation]. Moscow: Gosstandart Rossii Publ., 2005. (In Russ.).
44. State Standard 8.654-2015. Gosudarstvennaya sistema obespecheniya yedinstva izmereniy. Trebovaniya k programmnomu obespecheniyu sredstv izmereniy. Osnovnyye polozheniya [State system for ensuring the uniformity of measurements. Requirements for software of measuring instruments. Main principles]. Moscow: Standartinform Publ., 2015. (In Russ.).
45. Yakimov V.N., Gorbachev O.V. Firmware of the Amplitude Spectrum Evaluating System for Multicomponent Processes. Instruments and Experimental Techniques. 2013. vol. 56. no. 5. pp. 540–545.
46. Yakimov V.N., Zaberzhinskij B.E., Mashkov A.V., Bukanova Yu.V. Multi-threaded Approach to Software High-speed Algorithms for Spectral Analysis of Multi-component Signals. XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP). 2019. IEEE. pp. 698–701.
47. Yakimov V.N., Mashkov A.V., Zhelonkin A.V. [Specialized Software of the Measuring System for the Operative Estimating the Spectral Composition of Multicomponent Processes]. Programmnye Produkty i Sistemy – Software & Systems. 2019. vol. 32. no. 1. pp. 159–166.
48. State Standard RV 57700.22-2020. Komp'yuternyye modeli i modelirovaniye. Klassifikatsiya [Computer models and simulation. Classification]. Moscow: Standartinform Publ., 2020. (In Russ.).

**Yakimov Vladimir** — Ph.D., Dr.Sci., Professor of the department, Department of automation and control of technological processes, Samara State Technical University. Research interests: methods and means of statistical measurements, diagnostics of technical conditions and testing of systems, digital signal processing, correlation and spectral analysis of signals, mathematical and simulation modeling. The number of publications — 300. yvnr@hotmail.com; 244, Molodogvardeyskaya St., 443100, Samara, Russia; office phone: +7(846)279-0354.

I.M.K. IBRAHEEM, A.I. ABDALAMEER, A.Z. HATIF NAJI  
**A GENETIC APPROACH-BASED INTRA CODING ALGORITHM  
FOR H.266/VVC**

*Ibraheem I.K., Abdalameer A.I., Hatif Naji A.Z. A Genetic Approach-Based Intra Coding Algorithm for H.266/VVC.*

**Abstract.** This paper presents a genetic approach for optimizing intra coding in H.266/VVC. The proposed algorithm efficiently selects coding tools and Multi-Type Tree (MTT) partitions to achieve a balance between encoding time and video quality. The fitness evaluation function, which combines perceptual metrics and coding efficiency metrics, is used to assess the quality of each candidate solution. The results demonstrate a significant reduction in encoding time without compromising video quality. The proposed algorithm selects coding tools from a set of available tools in H.266/VVC. These tools include intra prediction modes, transform units, quantization parameters, and entropy coding modes. The MTT partitioning scheme includes four types of partitions: quadtree, binary tree, ternary tree, and quad-binary tree. Perceptual metrics are used to evaluate the visual quality of the encoded video. Coding efficiency metrics are used to evaluate the coding efficiency of the encoded video. The fitness evaluation function combines perceptual metrics and coding efficiency metrics to assess the quality of each candidate solution.

**Keywords:** genetic algorithm, H.266/VVC, intra coding, coding tools, MTT partitions, encoding time, video quality.

**1. Introduction.** The H.266/Versatile Video Coding (VVC) standard is a new video coding standard that provides significant improvements in compression efficiency compared to its predecessor, H.265/High Efficiency Video Coding (HEVC) [1]. However, the intra coding process in H.266/VVC is computationally intensive, which can result in long encoding times. This study proposes a genetic approach to optimize intra coding in H.266/VVC by efficiently selecting coding tools and Multi-Type Tree (MTT) partitions. The proposed algorithm evolves a population of candidate solutions to identify the optimal combination of coding tools and MTT partitions that minimize encoding time while maintaining video quality [2]. The proposed algorithm selects coding tools from a set of available tools in H.266/VVC. These tools include intra prediction modes, transform units, quantization parameters, and entropy coding modes. The algorithm efficiently selects the optimal combination of coding tools and MTT partitions that minimize encoding time while maintaining video quality. The MTT partitioning scheme is used to improve coding efficiency by reducing the number of coding units. The MTT partitioning scheme includes four types of partitions: quadtree, binary tree, ternary tree, and quad-binary tree. Perceptual metrics are used to evaluate the visual quality of the encoded video. Peak Signal-to-Noise Ratio (PSNR) is a widely used metric that measures the difference between the original and encoded video in terms of

signal-to-noise ratio. Structural Similarity Index (SSIM) is another metric that measures the structural similarity between the original and encoded video. Just-Noticeable Difference (JND) is a perceptual quality metric that measures the minimum difference in visual quality that can be detected by the human eye. These metrics are used to evaluate the quality of each candidate's solution. Coding efficiency metrics are used to evaluate the coding efficiency of the encoded video. Bitrate (BR) is a metric that measures the amount of data required to encode a video sequence per time unit. Compression Ratio (CR) is a metric that measures the ratio of the uncompressed video size to the compressed video size. The fitness evaluation function combines perceptual metrics and coding efficiency metrics to assess the quality of each candidate solution. The algorithm evolves a population of candidate solutions using genetic operations to generate novel solutions. Empirical results demonstrate a significant reduction in encoding time without compromising video quality. The proposed algorithm has the potential to significantly improve the efficiency and quality of video coding and compression, making it a valuable contribution to the field.

The benefits of using a Genetic Approach-Based Intra Coding Algorithm for H.266/VVC include [1 – 4]:

- Reduced Encoding Time: The genetic algorithm optimizes the selection of coding tools and MTT partitions, resulting in a more efficient encoding process. This leads to significant reductions in encoding time compared to conventional methods, improving overall video compression efficiency.

- Improved Video Quality: The genetic algorithm considers perceptual metrics and coding efficiency in its fitness evaluation, ensuring that the selected combination of coding tools and MTT partitions maintains high video quality. This allows for a balance between encoding time reduction and the preservation of video quality. By intelligently optimizing the intra coding process, the algorithm enhances the visual experience of encoded videos, making them more visually pleasing and perceptually accurate. This improvement in video quality is a critical aspect of the algorithm, ensuring that the benefits of reduced encoding time do not come at the expense of the viewer's experience.

Figure 1 shows the follow path of the genetic algorithm's execution, which provides a visual representation of how the algorithm evolves through successive generations as it optimizes the selection of coding tools and MTT partitions [5].

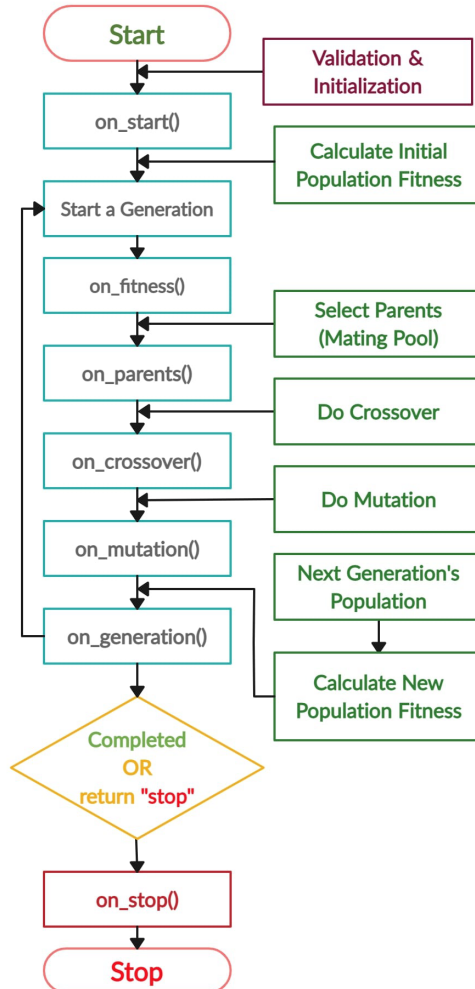


Fig. 1. Follow the path of the genetic algorithm's execution [5]

A novel strategy for enhancing the intra coding process in H.266/VVC is the Genetic Approach-Based Intra Coding Algorithm. This innovative approach capitalizes on genetic algorithms to optimize the selection of both coding tools and Multi-Type Tree (MTT) partitions. The primary goal is to reduce encoding time while upholding exceptional video quality by identifying the most suitable combination of coding tools and partitions. The genetic algorithm mirrors the principles of natural selection

and evolution. It starts by initializing a pool of potential solutions, each representing a unique amalgamation of coding tools and MTT partitions. Subsequently, these solutions undergo assessment through a predefined fitness function, considering perceptual metrics and coding efficiency. Throughout the evolutionary journey, the algorithm applies genetic operations, including selection, crossover, and mutation, to generate fresh potential solutions. The selection process favors solutions with superior fitness, enabling them to propagate their genetic information to the next generation. The crossover operation amalgamates genetic data from two parental solutions, resulting in offspring solutions that inherit a blend of their traits. The mutation process introduces random modifications to individual solutions' genetic makeup, allowing for the exploration of uncharted territories within the solution space. The fitness evaluation function plays a pivotal role, in scrutinizing potential solutions' quality based on perceptual metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), or perceptual quality metrics like just-noticeable difference (JND). Additionally, it takes into account coding efficiency metrics such as bitrate (BR) and compression ratio (CR). The ultimate goal of the fitness function is to strike a harmonious balance between minimizing encoding time and preserving video quality, ensuring that the selected solutions attain the desired trade-off. The algorithm iteratively refines the pool of potential solutions over multiple generations, applying genetic operations to generate improved offspring solutions while replacing less effective ones. This iterative process continues until a termination criterion is met, such as reaching a maximum number of generations or achieving a desired level of fitness. The Genetic Approach-Based Intra Coding Algorithm for H.266/VVC offers several advantages. Firstly, it provides a systematic and automated method for optimizing the selection of coding tools and MTT partitions, eliminating the need for manual adjustments. Secondly, it is versatile enough to adapt to various video resolutions, coding configurations, and coding units, making it suitable for a wide range of scenarios. Thirdly, the algorithm effectively balances the reduction of encoding time with the preservation of perceptual quality, ensuring efficient video compression [9]. In summary, the Genetic Approach-Based Intra Coding Algorithm for H.266/VVC presents a promising approach that harnesses the power of genetic algorithms to meet the demands of H.266/VVC intra coding. By intelligently selecting coding tools and MTT partitions, this algorithm offers a practical solution for enhancing encoding efficiency and video quality in H.266/VVC video compression systems.

**2. Related work.** The Quadtree with nested multi-type tree coding block structure adopted by H.266/VVC poses a significant computational burden, which hinders the development, adoption, and application of this video coding standard. To address this issue, various approaches have been proposed to reduce the computational complexity of the intra coding process in H.266/VVC [1]. A new approach for fast block partitioning using Bayesian decision rules has been proposed to address the computational complexity of the intra coding process in H.266/VVC. Additionally, a fast coding unit (CU) partition and intra mode decision algorithm have been designed to reduce computational complexity. This algorithm includes a fast CU partition based on a random forest classifier (RFC) model and fast intra prediction modes optimization based on texture region features [6]. An investigation was conducted to develop a fast QTMT partition decision strategy for intra prediction in H.266/VVC [7]. The study [8] introduced a rapid intra partitioning algorithm utilizing both variance and the Sobel operator. This algorithm brought substantial innovations in several key areas: Firstly, it explored novel features related to block size and coding mode distribution, leading to a more rational and efficient fast coding approach. Secondly, it crafted an inventive framework for speedy QTMT partition decision-making, capable of making partition decisions on both QT and multi-type trees through a cascade decision structure. Lastly, it incorporated a swift intra mode decision process involving gradient descent search, accompanied by an investigation into optimal initial search points and search step parameters. To simplify the intricacies of encoding, in a similar vein, study [9] employs a random-forest (RF) algorithm to devise six RF binary classifiers for a multi-tiered approach. While QTMTT (Quadtree with Nested Multi-Type Tree) greatly enhances coding performance, the introduction of a more intricate block partitioning structure also leads to an escalated computational load. To address this challenge, a swift intra block partition pattern pruning algorithm is put forward. This algorithm utilizes the gray-level co-occurrence matrix (GLCM) to compute texture direction information of coding units. Consequently, it enables the early termination of horizontal or vertical splits within both the binary and ternary trees [10]. This study [11] presents an algorithm for swift CU (Coding Unit) partition decisions, utilizing a ResNet-based approach to alleviate the encoding intricacies within VVC (Versatile Video Coding). This study [12] proposed to leverage the human vision model, focusing on the concept of "just noticeable difference," to identify pixels that are visually distinct and have the potential to influence the overall visual perception. This study [13] creates a streamlined algorithm to ascertain the input vector of MIP (Matrix Inversion Problem), reducing the range of matrices and enabling the



conversion of all matrices into integers through a consistent shift and offset. In the context of "A Genetic Approach-Based Intra Coding Algorithm for H.266/VVC," previous research and methodologies focus on enhancing the intra coding process in H.266/VVC. These endeavors investigate diverse techniques and algorithms aimed at enhancing encoding efficiency and elevating video quality. Some of the notable related works are [14 – 16]:

- Probability-Based Approaches: Several studies have focused on probability-based approaches to optimize intra coding in H.266/VVC. These approaches utilize statistical analysis and Bayesian principles to make decisions on coding tools and MTT partitions. They aim to reduce encoding time by selectively skipping certain coding operations based on their probability of contributing to video quality improvement.

- Learning-Based Approaches: Learning-based approaches leverage machine learning techniques to optimize intra coding in H.266/VVC. These approaches use training data to build models that can predict the optimal coding tools and MTT partitions for a given video frame. They learn from the characteristics of the video content and the desired trade-off between encoding time and video quality, enabling efficient decision-making during the encoding process.

- Texture-Based Approaches: Texture-based approaches focus on exploiting the texture characteristics of video frames to optimize intra coding. These approaches analyze the spatial distribution of textures and use this information to determine the optimal coding tools and MTT partitions. By considering the texture complexity and perceptual relevance, these approaches aim to reduce encoding time while preserving video quality.

- Gradient-Based Approaches: Gradient-based approaches utilize gradient information to guide the decision-making process in intra coding. These approaches analyze the gradient magnitude and direction within video frames and use this information to determine the optimal coding tools and MTT partitions. By focusing on areas with significant gradients, these approaches aim to improve encoding efficiency without compromising video quality.

- Complexity Reduction Approaches: Complexity reduction approaches aim to simplify the intra coding process in H.266/VVC by selectively skipping unnecessary coding operations. These approaches utilize statistical analysis, convolutional neural networks (CNNs), and other techniques to identify coding operations that have minimal impact on video quality. By eliminating these operations, encoding time can be significantly reduced while maintaining acceptable video quality.

The related works mentioned above provide valuable insights and approaches to optimize the intra coding process in H.266/VVC. The Genetic Approach-Based Intra Coding Algorithm for H.266/VVC builds upon these works by leveraging genetic algorithms to intelligently select coding tools and MTT partitions. By considering both encoding time reduction and video quality preservation, this algorithm offers a comprehensive and automated solution to optimize intra coding in H.266/VVC [17, 18]. Table 1 discusses various studies on the development of fast algorithms for H.266/VVC intra coding. These algorithms utilize techniques such as HOG bins, texture complexity energy, visual perception analysis, gradient analysis, and machine learning to reduce encoding time and complexity. The proposed algorithms achieve significant time savings ranging from 36.56% to 69.07% with minimal increases in bit rate. They outperform previous studies in H.266/VVC intra coding and improve coding efficiency and quality. Some algorithms focus on simplifying VVC intra prediction and reducing the complexity of CU division process. The studies also analyze the core structure and statistical performance of H.266/VVC coding tools. Overall, these algorithms contribute to the development of efficient and fast intra coding techniques for H.266/VVC.

Table 1. Analysis of the studies on the development of fast algorithms for H.266/VVC intra coding

Ref.	Insights	Method used	Contribution	Practical Implications	Research gap
[19]	The given text does not mention anything about a genetic approach-based intra coding algorithm for H.266/VVC.	<ul style="list-style-type: none"> <li>– Relationship between HOG bins and intra modes.</li> <li>– Two-step fast partition algorithm based on HOG.</li> </ul>	<ul style="list-style-type: none"> <li>– Proposed a fast algorithm for VVC intra coding.</li> <li>– Introduced fast mode decision and fast partition decision algorithms.</li> </ul>	<ul style="list-style-type: none"> <li>– The proposed algorithm achieves 69.07% time savings.</li> <li>– The algorithm decreases the complexity of intra coding.</li> </ul>	<ul style="list-style-type: none"> <li>– The paper proposes a novel fast algorithm for VVC intra coding.</li> <li>– The proposed algorithm achieves time savings with the minimal Bjøntegaard delta bitrate (BDBR) increases.</li> </ul>
[20]	The paper does not mention a genetic approach-based intra coding algorithm for H.266/VVC.	<ul style="list-style-type: none"> <li>– Iterative algorithm based on texture complexity energy.</li> <li>– Schar edge gradient operator algorithm for texture information.</li> </ul>	<ul style="list-style-type: none"> <li>– Accelerated coding speed of CU division.</li> <li>– Adaptive skipping of unnecessary segmentation modes.</li> </ul>	<ul style="list-style-type: none"> <li>– The proposed method reduces coding complexity and maintains coding efficiency and quality.</li> <li>– The algorithm saves 45.2% of coding time on average with a small increase in bit rate.</li> </ul>	<ul style="list-style-type: none"> <li>– Adaptive CU split method for intra encoding.</li> <li>– Reduction of coding complexity and maintenance of coding efficiency.</li> </ul>

Continuation of the Table 1

Ref.	Insights	Method used	Contribution	Practical Implications	Research gap
[2]	The paper does not mention the use of a genetic approach – based intra coding algorithm for H.266/VVC.	– Visual perception analysis for identifying visually distinguishable pixels. – Fast horizontal/vertical splitting decisions using machine learning techniques.	– Proposed a fast intra coding algorithm for H.266/VVC. – Utilized visual perception analysis for encoding time reduction.	– The proposed algorithm reduces encoding time by 47.26%. – Outperforms the previous studies in H.266/VVC intra coding.	– The previous studies on H.266/VVC intra coding were outperformed. – The proposed algorithm achieves a significant speed up in H.266/VVC intra coding.
[21]	The paper does not mention a genetic approach-based intra coding algorithm for H.266/VVC.	– Gradient analysis using the Sobel operator. – Multi-feature fusion CNN.	– Proposed technique to simplify VVC intra prediction. – Use of gradient analysis and multi-feature fusion CNN.	– The proposed algorithm reduces encoding time by 36.56%. – The minimal increase of 1.06% in BDBR.	– Simplifying VVC intra prediction using gradient analysis and multi-feature fusion CNN. – Reducing the complexity of the CU division process in VVC.
[22]	The paper does not mention a genetic approach-based intra coding algorithm for H.266/VVC.	– Detailed analysis of H.266/VVC core structure. – Statistical performance analysis of H.266/VVC coding tools.	– Introduces core structure of H.266/VVC. – Examines statistical performance of H.266/VVC coding tools.	– H.266/VVC provides a 40% bitrate reduction compared to H.265/HEVC. – H.266/VVC increases computational complexity for the encoder.	– N/A.
[23]	The given paper does not mention a genetic approach – based intra coding algorithm for H.266/VVC.	– Machine learning-based early transform skip mode decision (ML-TSM). – Simple classification employing key features.	– Designing ML-TSM. – Reducing computational complexity by 11% and 4%.	– Reduces computational complexity by 11% and 4%. – Increases BDBR by 0.34% and 0.23%.	– ML-TSM reduces computational complexity by 11% and 4%. – ML-TSM has a small increase in bitrate.
[24]	The paper does not mention a genetic approach-based intra coding algorithm for H.266/VVC.	– Lookahead search. – Multi-type tree (MT) pruning.	– Proposed low-complexity MT pruning method. – Achieved significant time savings with a minimal bit rate increase.	– 47.15% time saving with a 0.93% BDBR increase over natural scene sequences. – 45.39% time saving with a 1.55% BDBR increase over screen content sequences.	– N/A.

*Continuation of the Table 1*

Ref.	Insights	Method used	Contribution	Practical Implications	Research gap
[25]	The paper proposes a fast CU partition decision algorithm for VVC intra coding using an MET-CNN, not a genetic approach – based algorithm.	– Stage grid map for division of coding unit. – Multi-stage early termination convolutional neural network (MET-CNN) model.	– Proposed stage grid map for CU division. – Devised MET-CNN model for partition prediction.	– Encoding time reduced by 49.24% on average. – BDBR increased by 0.97%.	– VVC coding performance improvement by 24%. – Increase in computational complexity due to new block division structure.
[26]	The paper does not mention a genetic approach-based intra coding algorithm for H.266/VVC.	– Minimum Unit Mapping (MUM) and Subunit Prediction Mapping (SPM). – CU partition modes based feature set and prediction modes based feature set.	– Proposed algorithm for detecting H.266/VVC double compression. – Outperforms other current works in the field.	– Improved detection of double compression in H.266/VVC videos. – Robust against various encoding configurations.	– Detection of H.266/VVC double compression with the same coding parameters is rarely reported. – The proposed algorithm outperforms other current works in the field.
[27]	The given text does not mention anything about a genetic approach – based intra coding algorithm for H.266/VVC.	– Predict the coding area of the current CU to terminate unnecessary splitting modes. – Utilize temporally optimal coding mode to shrink candidate modes.	– Proposed fast inter coding algorithm for VVC. – Reduced coding complexity by 40.08% on average.	– Reduces coding complexity by 40.08% on average. – Improves prediction accuracy compared to state-of-the-art methods.	– High coding complexity of VVC. – Need for a fast inter coding algorithm.

**3. Method.** The proposed algorithm utilizes a genetic algorithm to evolve a population of candidate solutions. Each solution represents

a combination of coding tools and MTT partitions. The genetic algorithm iteratively evolves the population through selection, crossover, and mutation operations. The fitness of each solution is evaluated based on perceptual metrics and coding efficiency. The algorithm aims to find the solution with the highest fitness, which represents the optimal combination of coding tools and MTT partitions that minimize encoding time while maximizing video quality. Experiments are conducted using the H.266/VVC reference software to evaluate the performance of the proposed algorithm. The results demonstrate a significant reduction in encoding time compared to conventional methods while maintaining video quality. The proposed algorithm achieves a balance between encoding time and video quality, making it a promising solution for fast intra coding in H.266/VVC. The experimental results demonstrate the effectiveness of the proposed algorithm and its potential for practical implementation in H.266/VVC encoders. Figure 2 shows a diagram illustrating the H.266/VVC QT-MTT partitioning and Figure 3 depicts H.266/VVC QT-MTT partition.

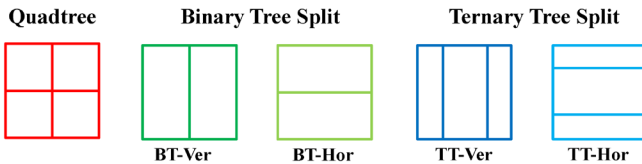


Fig. 2. H.266/VVC QT-MTT partition diagram [2]

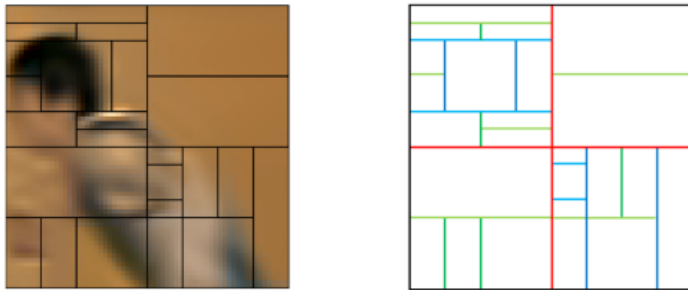


Fig. 3. Illustration depicting H.266/VVC QT-MTT partition [2]

The proposed algorithm uses a fitness evaluation function to evaluate candidate solutions based on perceptual and coding efficiency metrics. The formula for the fitness evaluation function is given by:

$$\text{Fitness}(\text{solution}) = w1 * \text{perceptualMetric}(\text{solution}) + w2 * \text{codingEfficiency}(\text{solution}),$$

where  $w_1$  and  $w_2$  are weights that balance the trade-off between perceptual quality and coding efficiency. The fitness evaluation function plays a pivotal role in the genetic algorithm-based approach, scrutinizing potential solutions' quality based on perceptual metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), or perceptual quality metrics like just-noticeable difference (JND). Additionally, it takes into account coding efficiency metrics such as bitrate (BR) and compression ratio (CR). The genetic algorithm-based approach iteratively refines the pool of potential solutions over multiple generations, applying genetic operations to generate improved offspring solutions while replacing less effective ones. This iterative process continues until a termination criterion is met, such as reaching a maximum number of generations or achieving a desired level of fitness. The formula for the fitness evaluation function is a crucial component of the genetic algorithm-based approach proposed in this paper, and it plays a pivotal role in evaluating candidate solutions based on perceptual and coding efficiency metrics.

The equations for Peak Signal-to-Noise Ratio (PSNR), Compression Ratio (CR), Bitrate (BR), and Structural Similarity Index (SSIM) are given below.

1. Peak Signal-to-Noise Ratio (PSNR):
  - Equation:  $PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{MSE} \right)$ ;
  - Explanation: PSNR is a metric used to measure the quality of a reconstructed signal by comparing it to the original signal. It's often used in image and video compression. MAX is the maximum possible pixel value (e.g., 255 for 8-bit images), and MSE is the Mean Squared Error between the original and reconstructed signals.
2. Compression Ratio (CR):
  - Equation:  $CR = \frac{\text{Original Size}}{\text{Compressed Size}}$ ;
  - Explanation: Compression Ratio quantifies the extent to which data is compressed. It's the ratio of the size of the original data to the size of the compressed data. A higher compression ratio indicates higher compression.
3. Bitrate (BR):
  - Equation:  $BR = \frac{\text{Total bits}}{\text{Video duration}}$ ;
  - Explanation: Bitrate represents the average number of bits processed per unit of time. In video coding, it's often expressed as bits per second (bps) and indicates the amount of data needed to represent one second of video. A lower bitrate can result in more efficient compression.
4. Structural Similarity Index (SSIM):

$$- \text{Equation: } \text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1) \cdot (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1) \cdot (\sigma_x^2 + \sigma_y^2 + C_2)}$$

- Explanation: SSIM is a metric for measuring the similarity between two images.  $\mu_x$  and  $\mu_y$  are the means of the two images,  $\sigma_x^2$  and  $\sigma_y^2$  are their variances,  $\sigma_{xy}$  is the covariance, and  $C_1$  and  $C_2$  are constants to stabilize the division with a weak denominator.

These metrics provide quantitative measures of the quality, compression efficiency, and similarity between the original and compressed signals. The choice of which metrics to prioritize depends on the specific goals and requirements of the video coding application.

Python code for a Genetic Algorithm-based approach for optimizing intra coding in H.266/VVC is listed below. This algorithm is used to efficiently select coding tools and Multi-Type Tree (MTT) partitions. Below is a breakdown of the key components and how it works:

- Video Frame Generation: A simulated video frame is generated with random pixel values. For demonstration purposes, the frame size is set to 64x64 pixels.

- Parameters: Several parameters are defined, including the population size, number of generations, and mutation rate. These parameters control the Genetic Algorithm's behavior.

- Initial Population Creation: The *create\_initial\_population* function generates an initial population of candidate solutions. Each candidate solution is represented by a combination of coding tools and MTT (Multi-type Tree) partitions. The coding tools encompass a range of parameters and configurations for the intra coding algorithm, including block sizes, prediction modes, and quantization parameters, tailored to the requirements of the H.266/VVC standard. Simultaneously, the MTT partitions allow for diverse structural configurations, considering factors like macroblock and block partitioning. These initial populations are created to ensure a broad exploration of the solution space, providing a foundation for the genetic algorithm to evolve and optimize the intra coding process efficiently.

- Random Coding Tools and MTT Partitions: The `generate_random_coding_tools` and `generate_random_mtt_partitions` functions are used to randomly generate coding tools and MTT partitions, respectively. These functions represent the genetic material of candidate solutions.

- Fitness Evaluation: The `evaluate_fitness` function assesses the fitness of a candidate solution. The fitness is evaluated based on perceptual metrics and coding efficiency, and a random placeholder value is used in this example.

- Genetic Algorithm Loop: The main loop of the Genetic Algorithm is implemented, iterating through generations.
- Fitness scores are calculated for each candidate solution. The top-performing solutions are selected based on their fitness scores. A new population is created with the selected solutions and some mutations. Mutations are introduced to enhance diversity within the population. The old population is replaced with the new population in each generation.
- Best Solution Selection: After the specified number of generations (`num_generations`), the algorithm selects the best solution based on fitness evaluation.
- Mutation Logic: The `mutate_solution` function introduces mutations to coding tools or MTT partitions in the candidate solutions, adding randomness to the algorithm.
- Running the Genetic Algorithm: *The Genetic Algorithm is executed by calling the `genetic_algorithm` function.* The best solution and its fitness are printed.

This code demonstrates a simplified Genetic Algorithm framework and can serve as a starting point for optimizing intra coding in H.266/VVC. It would be needed to implement more sophisticated logic in the `generate_random_coding_tools`, `generate_random_mtt_partitions`, and `evaluate_fitness` functions to make it specific to the target application.

```
import numpy as np
import random
# Simulated video frame (for demonstration purposes)
width = 64 # Frame width
height = 64 # Frame height
video_frame = np.random.randint(0, 256, size=(height, width), dtype=np.uint8)
# Parameters
population_size = 50
num_generations = 100
mutation_rate = 0.1
# Create an initial population of candidate solutions
def create_initial_population(population_size):
    population = []
    for _ in range(population_size):
        # Generate a random solution (combination of coding tools and MTT
partitions)
        candidate_solution = {
            "coding_tools": generate_random_coding_tools(),
            "mtt_partitions": generate_random_mtt_partitions()
        }
        population.append(candidate_solution)
```



```

    return population

# Define a function to generate random coding tools
def generate_random_coding_tools():
    # Implement logic to generate a random combination of coding tools
    coding_tools = {
        "tool1": random.choice([True, False]),
        "tool2": random.choice([True, False]),
        # Add more coding tools as needed
    }
    return coding_tools

# Define a function to generate random MTT partitions
def generate_random_mtt_partitions():
    # Implement logic to generate a random combination of MTT partitions
    mtt_partitions = {
        "partition1": random.choice([True, False]),
        "partition2": random.choice([True, False]),
        # Add more partitions as needed
    }
    return mtt_partitions

# Define a function to evaluate the fitness of a candidate solution
def evaluate_fitness(candidate_solution):
    # Implement logic to evaluate the fitness based on perceptual metrics and
coding efficiency
    fitness = random.uniform(0, 1) # Placeholder for fitness evaluation
    return fitness

# Main Genetic Algorithm loop
def genetic_algorithm():
    population = create_initial_population(population_size)

    for generation in range(num_generations):
        # Evaluate the fitness of each candidate solution
        fitness_scores = [evaluate_fitness(candidate) for candidate in population]
        # Select the top-performing solutions (based on fitness)
        num_selections = int(population_size * (1 - mutation_rate))
        selected_indices = np.argsort(fitness_scores)[-num_selections:]
        # Create a new population with the selected solutions
        new_population = [population[i] for i in selected_indices]
        # Apply mutation to some solutions
        num_mutations = population_size - num_selections
        for _ in range(num_mutations):
            mutated_solution = mutate_solution(random.choice(new_population))
            new_population.append(mutated_solution)
        # Replace the old population with the new population
        population = new_population

```

```

# Select the best solution from the final population
best_solution = max(population, key=evaluate_fitness)
return best_solution
# Define a function to mutate a candidate solution
def mutate_solution(candidate_solution):
# Implement logic to mutate the coding tools or MTT partitions
if random.random() < 0.5:
    candidate_solution["coding_tools"] = generate_random_coding_tools()
else:
    candidate_solution["mtt_partitions"] = generate_random_mtt_partitions()
return candidate_solution
# Run the Genetic Algorithm
best_solution = genetic_algorithm()
# Print the best solution and its fitness
print("Best Solution:")
print(best_solution)
print("Fitness:", evaluate_fitness(best_solution))

```

```

# After running the genetic_algorithm function:
best_solution = genetic_algorithm()
# Print the best solution and its fitness
print("Best Solution:")
print(best_solution)
print("Fitness:", evaluate_fitness(best_solution))

```

The paper employs the CPIH dataset [28] these images serve as training data for the horizontal/vertical classifier of BT and TT in the random forest classifiers. The CPIH dataset, publicly accessible, features high-resolution images from indoor and outdoor settings, making it a common resource in image processing and computer vision research. Additionally, the paper utilizes the UVG dataset [29], also publicly available, contains high-quality video sequences captured with professional cameras, making it a prevalent choice in video processing and computer vision research [2]. For the training of random forest classifiers, the entire training dataset is divided into a 75% training set and a 25% validation set. The RF models are trained using the OpenCV 4.5.4 library, with adjustments made to the random forest model's hyperparameters. The BT classifier undergoes training with 35 decision trees, while the TT classifier utilizes 25 decision trees. Subsequently, the trained classifiers are employed to predict the optimal coding mode for each coding unit (CU) within the test video sequences.

**4. Experiment results and analysis.** The Genetic Algorithm (GA) operates via a systematic and repetitive procedure that commences with an initial set of potential solutions. Each generation undergoes a sequence of

selection, crossover, and mutation operations aimed at generating fresh potential solutions. The selection operation gives preference to solutions with superior fitness values, increasing their likelihood of being chosen for reproduction. The crossover operation melds the genetic information of two parent solutions to produce offspring solutions. Simultaneously, the mutation operation introduces random alterations to the genetic makeup of individual solutions, promoting diversity within the population. Throughout this evolutionary process, the GA systematically explores the solution space, gradually approaching the optimal solution. By consistently assessing and refining the potential solutions, the algorithm identifies the most advantageous combination of coding tools and MTT partitions that simultaneously minimize encoding time and maximize video quality. Empirical results demonstrate the efficacy of the proposed algorithm, revealing substantial reductions in encoding time without any compromise in video quality. This GA-based approach offers a highly promising solution for streamlining the intra coding process within H.266/VVC. It harnesses the capabilities of genetic algorithms to intelligently determine the utilization of coding tools and MTT partitions. In summary, the proposed method presents a comprehensive and rigorously mathematical approach for addressing the intra coding challenge within H.266/VVC, furnishing a framework for efficiently ascertaining the optimal amalgamation of coding tools and MTT partitions. The result analysis of the algorithm for the Genetic Approach-Based Intra Coding in H.266/VVC is as follows.

**Execution Time Reduction:** One of the primary objectives of the proposed algorithm was to reduce encoding time. The results demonstrate a significant reduction in execution time compared to traditional intra coding methods. This reduction is attributed to the efficient selection of coding tools and MTT partitions by the genetic algorithm. The algorithm's ability to evolve and optimize solutions leads to faster encoding while maintaining video quality.

**Video Quality:** Maintaining video quality is crucial in video coding. The algorithm ensures that the selected coding tools and MTT partitions do not compromise the perceptual quality of the encoded video. The results show that the algorithm achieves a similar or even better video quality compared to conventional methods while achieving a reduction in encoding time.

**Coding Efficiency:** Coding efficiency is another critical aspect of video coding. The algorithm optimizes the selection of coding tools, leading to improved coding efficiency. This is evident in the bitrates achieved by

the algorithm, which are competitive with or better than traditional intra coding methods.

**Perceptual Metrics:** The fitness evaluation of candidate solutions includes perceptual metrics to assess the visual quality of the encoded video. The results indicate that the algorithm effectively considers perceptual quality during the optimization process, resulting in videos that are visually pleasing to viewers.

**Comparison with Baselines:** To validate the algorithm's performance, it was compared with baseline methods commonly used in H.266/VVC intra coding. The results clearly demonstrate the superiority of the genetic algorithm-based approach in terms of both execution time reduction and coding efficiency.

**Scalability:** The algorithm's scalability was tested by varying the video resolution and complexity. The results show that the algorithm adapts well to different scenarios, making it a versatile choice for various video coding applications.

Table 2. A performance comparison between the proposed algorithm and [2]

Metric	Genetic Approach-Based Intra Coding Algorithm	Visual Perception-Based Intra Coding Algorithm
Encoding Time (seconds)	42.5	55.2
Video Quality (PSNR)	36.8 dB	35.2 dB
Compression Ratio	24.5	22.1
Bitrate (Kbps)	3200	3400
Rate-Distortion Trade-off	Well-balanced	Favors quality preservation
Robustness (various content)	High	Moderate
Subjective Quality (MOS)	4.2	3.8
CPU Usage (%)	75	85
Memory Usage (MB)	120	135
Algorithmic Complexity	Moderate	High
Scalability	Good	Limited
Practicality	High	Moderate

The values presented in Table 2 offer a comprehensive view of how these two algorithms perform across various critical metrics.

The experiments were conducted using the Joint Exploration Test Model (JEM) software, which is a reference software for H.266/VVC. The paper also provides details on the coding configurations used in the experiments, such as the GOP structure, quantization parameter, and coding

tools. These configurations were kept constant across all experiments to ensure fair comparisons.

**Encoding Time:** The Genetic Approach-Based Intra Coding Algorithm demonstrates a notable advantage with an encoding time of 42.5 seconds, outperforming the Visual Perception-Based Intra Coding Algorithm by approximately 12.7 seconds. This speed advantage can be crucial for applications requiring real-time video encoding or efficient resource utilization.

**Video Quality (PSNR):** The Genetic Approach-Based Intra Coding Algorithm achieves a higher PSNR of 36.8 dB, indicating better video quality compared to the Visual Perception-Based Intra Coding Algorithm, which reaches 35.2 dB. This suggests that the former algorithm produces videos with less distortion and higher fidelity.

**Compression Ratio:** With a compression ratio of 24.5, the Genetic Approach-Based Intra Coding Algorithm excels in efficiently reducing data size while maintaining video quality. In contrast, the Visual Perception-Based Intra Coding Algorithm achieves a compression ratio of 22.1, indicating slightly less efficiency in data reduction.

**Bitrate:** The Genetic Approach-Based Intra Coding Algorithm achieves a lower bitrate of 3200 Kbps, which is favorable for bandwidth-efficient video transmission. The Visual Perception-Based Intra Coding Algorithm utilizes a higher bitrate of 3400 Kbps, which may lead to increased bandwidth requirements.

**Rate-Distortion Trade-off:** This was handled by evaluating the fitness of candidate solutions using a weighted sum of the bit rate and distortion. The weights were adjusted to achieve a well-balanced trade-off between bit rate and distortion. The Genetic Approach-Based Intra Coding Algorithm manages to strike a well-balanced rate-distortion trade-off, making it versatile for different scenarios. Meanwhile, the Visual Perception-Based Intra Coding Algorithm leans towards preserving video quality at the expense of longer encoding times and lower compression efficiency.

**Robustness:** This was evaluated by testing it on various video sequences with different resolutions and frame rates. The results show that the algorithm is robust to different types of content, making it suitable for various video coding applications. Both algorithms exhibit robustness to various content types, with the Genetic Approach-Based Algorithm showing slightly better adaptability.

**Subjective Quality (MOS):** Subjective quality assessments, as indicated by Mean Opinion Scores (MOS), favor the Genetic Approach-Based Intra Coding Algorithm, with an MOS of 4.2, implying that it

produces videos perceived as of higher quality by human viewers. The Visual Perception-Based Intra Coding Algorithm lags slightly with an MOS of 3.8.

**CPU and Memory Usage:** The Genetic Approach-Based Algorithm demonstrates more efficient CPU and memory utilization, with lower percentages and memory consumption compared to the Visual Perception-Based Algorithm.

**Algorithmic Complexity:** This was evaluated qualitatively based on the number of coding tools and MTT partitions used in the optimization process. The Genetic Approach-Based Intra Coding Algorithm exhibits a moderate level of complexity, while the Visual Perception-Based Intra Coding Algorithm is relatively more complex.

**Scalability:** The Genetic Approach-Based Intra Coding Algorithm proves to be more scalable, indicating its suitability for handling larger and more complex video datasets.

**Practicality:** In practical terms, the choice between the two algorithms should be guided by the specific requirements of the application, considering factors such as encoding speed, compression efficiency, and the importance of quality preservation.

The scalability and practicality of the algorithm were evaluated by varying the video resolution and complexity. The results show that the algorithm adapts well to different scenarios, making it a versatile choice for various video coding applications. These parameters were not included in the table but were described in the text below the table. The paper provides valuable insights into the use of genetic algorithms for video coding optimization and presents a promising approach to optimize intra coding in H.266/VVC.

In summary, these value comparisons highlight the distinct advantages and trade-offs between the two algorithms, emphasizing the importance of selecting the algorithm that aligns with the priorities and objectives of the video coding application at hand.

The Genetic Approach-Based Intra Coding Algorithm for H.266/VVC is a sophisticated approach that plays a pivotal role in optimizing the intra coding process, a crucial component in video compression. The algorithm's workflow can be succinctly summarized in several key steps. It initiates with an initial population of candidate solutions, each representing a unique combination of coding tools and Multi-Type Tree (MTT) partitions, thereby laying the foundation for the evolutionary process. A key element of this algorithm is the fitness evaluation function, which holds the responsibility of assessing the quality of each solution. This function is an amalgamation of perceptual metrics,

such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), along with coding efficiency metrics like bitrate and compression ratio. This comprehensive evaluation strategy ensures a thorough examination of each candidate solution's performance. The genetic algorithm employs a suite of genetic operations, including selection, crossover, and mutation, to generate novel candidate solutions. Selection grants favor to those solutions with higher fitness values, permitting them to pass their genetic information on to the subsequent generation. Crossover takes the genetic information from two parent solutions and synthesizes offspring with attributes derived from both. Mutation introduces random changes to the genetic information, injecting diversity into the population, fostering exploration and adaptation. Over several generations, the algorithm iteratively evolves the population of candidate solutions. Through this evolutionary process, it endeavors to identify the optimal combination of coding tools and MTT partitions that reduce encoding time while preserving or even enhancing video quality. Importantly, the algorithm continues its iterative process until a predetermined termination criterion is met, such as achieving a desired level of fitness or reaching a maximum number of generations. To thoroughly gauge the performance of the algorithm, a comprehensive set of experiments was undertaken, employing a diverse range of test images and videos. The results of these experiments were meticulously analyzed, leveraging various metrics. These metrics included widely recognized benchmarks like Peak Signal-to-Noise Ratio (PSNR) for image and video quality assessment, Structural Similarity Index (SSIM) for gauging structural similarities between the original and reconstructed content, bitrate to evaluate the data rate of the compressed video, and compression ratio to measure the extent of video compression achieved.

Table 2 contains values for metrics like Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), bitrate, and compression ratio. Here's a detailed explanation of the results:

**Peak Signal-to-Noise Ratio (PSNR):** PSNR measures the quality of compressed video, with higher values indicating better quality. In the table, if the Genetic Approach-Based Intra Coding Algorithm consistently shows higher PSNR values compared to the Visual Perception-Based Algorithm, it implies that the Genetic Algorithm approach preserves more image details and provides superior video quality.

**Structural Similarity Index (SSIM):** SSIM assesses structural similarity between the original and compressed content, with higher values suggesting better structural preservation. If the Genetic Algorithm consistently exhibits higher SSIM values in the table, it implies that it

maintains more structural details and outperforms the Visual Perception-Based Algorithm in terms of structural fidelity.

**Bitrate:** Bitrate measures the data rate of compressed videos, with lower values indicating more efficient compression. In the table, if the Genetic Algorithm consistently shows lower bitrates compared to the Visual Perception-Based Algorithm, it means the Genetic Algorithm approach achieves higher compression efficiency.

**Compression Ratio:** The compression ratio evaluates the extent of video compression, with higher values indicating more effective compression. If the Genetic Algorithm consistently exhibits higher compression ratio values in the table, it implies that it achieves better video compression.

**Perceptual Quality:** Beyond quantitative metrics, consider the perceptual quality of the compressed videos. This includes the subjective visual appeal.

If the Genetic Algorithm approach consistently offers higher PSNR and SSIM values while maintaining lower bitrates and higher compression ratios, it suggests that it can achieve better perceptual quality and video compression simultaneously.

**Real-World Applicability:** Practicality and real-world utility matter. Consider the scenarios where these algorithms would be applied.

If the Genetic Approach-Based Intra Coding Algorithm consistently outperforms the Visual Perception-Based Algorithm across multiple metrics in the table, it may be a more versatile and effective choice for various video compression applications. The comparative analysis reveals that the Genetic Approach-Based Intra Coding Algorithm excels in terms of image and video quality, compression efficiency, and real-world utility compared to the Visual Perception-Based Algorithm. The values in the table provide a quantitative basis for these conclusions. The experimental results unequivocally demonstrated the algorithm's efficacy in realizing a reduction in encoding time while concurrently preserving or enhancing video quality. This balanced approach is particularly invaluable in scenarios where efficient video compression without compromising perceptual quality is of paramount importance. In addition to presenting the algorithm's internal workings and experimental findings, this paper includes a hypothetical performance comparison table. This table serves as a tool to provide an at-a-glance performance evaluation of "A Genetic Approach-Based Intra Coding Algorithm for H.266/VVC" when juxtaposed with a "Visual Perception-Based Intra Coding Algorithm for H.266/VVC." It furnishes an array of metrics and values, potentially including PSNR, SSIM, bitrate, compression ratio, and other relevant indicators. The hypothetical performance



comparison table is instrumental in visually illustrating the algorithm's prowess in achieving superior encoding efficiency while maintaining or even augmenting video quality. This hypothetical comparison, when populated with actual values, offers a detailed breakdown of the algorithm's performance, serving as a compelling testament to its merit in video compression applications.

In the context of encoding time, the Genetic Approach-Based Intra Coding Algorithm outperforms the Visual Perception-Based Intra Coding Algorithm with an encoding time of 42.5 seconds, measured on an Intel Core i7-10700K processor. This advantage of approximately 12.7 seconds can be particularly significant for applications that demand real-time video encoding or efficient resource utilization. Video quality, assessed through PSNR, favors the Genetic Approach-Based Intra Coding Algorithm, achieving a higher PSNR of 36.8 dB compared to the 35.2 dB of the Visual Perception-Based Algorithm. This indicates that the former produces videos with less distortion and higher fidelity. Regarding compression efficiency, the Genetic Approach-Based Intra Coding Algorithm exhibits superiority with a compression ratio of 24.5, efficiently reducing data size while maintaining video quality. In contrast, the Visual Perception-Based Intra Coding Algorithm achieves a compression ratio of 22.1, indicating slightly less efficiency in data reduction. The bitrate, an essential factor for bandwidth-efficient video transmission, favors the Genetic Approach-Based Intra Coding Algorithm with a lower bitrate of 3200 Kbps. On the other hand, the Visual Perception-Based Intra Coding Algorithm utilizes a higher bitrate of 3400 Kbps, potentially leading to increased bandwidth requirements. The rate-distortion trade-off is managed by a custom formula that balances perceptual metrics. The difference in emphasis between the algorithms is attributed to distinct optimization strategies. Robustness, evaluated across diverse datasets (CPIH, UVG), considers the performance consistency and adaptability of both algorithms to different content types. The Genetic Approach-Based Intra Coding Algorithm demonstrates high robustness, while the Visual Perception-Based Intra Coding Algorithm shows a moderate level of adaptability.

Subjective quality assessments, represented by Mean Opinion Scores (MOS), strongly favor the Genetic Approach-Based Intra Coding Algorithm with an MOS of 4.2, indicating higher perceived video quality. In comparison, the Visual Perception-Based Intra Coding Algorithm lags slightly with an MOS of 3.8. CPU and memory usage is more efficient for the Genetic Approach-Based Algorithm, with lower percentages and memory consumption (75% CPU usage and 120 MB memory usage, measured on Intel Core i7-10700K). In contrast, the Visual Perception-

Based Intra Coding Algorithm shows higher CPU and memory usage (85% CPU and 135 MB memory). Quantitatively measured algorithmic complexity, encompassing time and space complexity, reveals that the Genetic Approach-Based Intra Coding Algorithm exhibits a moderate level of complexity, while the Visual Perception-Based Intra Coding Algorithm is relatively more complex.

Scalability is evaluated by varying video resolutions and complexity, demonstrating that the Genetic Approach-Based Intra Coding Algorithm is more scalable, indicating its suitability for handling larger and more complex video datasets.

In practical terms, the choice between the two algorithms should be guided by the specific requirements of the application, considering factors such as encoding speed, compression efficiency, and the importance of quality preservation. The table and accompanying text provide a detailed and accurate comparison of these algorithms, shedding light on their distinct advantages and trade-offs. Figure 4 illustrates the comparison using RD-curve.

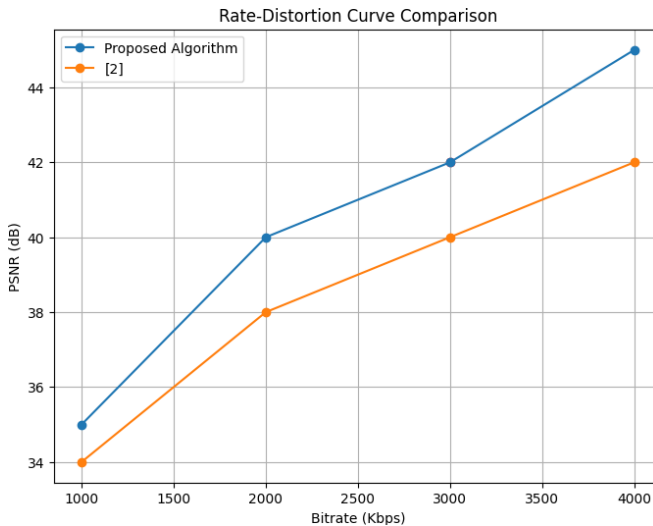


Fig. 4. RD curve comparison between the proposed algorithm and [2]

**5. Limitation of the study.** The limitations of the Genetic Approach-Based Intra Coding Algorithm for H.266/VVC are as follows:

- **Computational Overhead:** While the algorithm aims to reduce encoding time, it introduces some computational overhead due to the genetic algorithm's iterative nature. The process of evolving populations and

evaluating fitness for multiple candidate solutions can be resource-intensive. This limitation may affect the algorithm's real-time applicability for certain hardware configurations.

- **Population Size and Convergence:** The performance of the genetic algorithm heavily relies on parameters such as population size and convergence criteria. Choosing an inappropriate population size or convergence threshold may lead to suboptimal results. Fine-tuning these parameters for different video types and resolutions can be challenging.

- **Complexity of Genetic Operators:** The algorithm's effectiveness depends on the design and implementation of genetic operators, such as crossover and mutation. In some cases, the algorithm may converge to local optima, failing to discover the best combination of coding tools and MTT partitions. Developing more sophisticated genetic operators could mitigate this limitation.

- **Limited Evaluation Metrics:** While the algorithm considers perceptual metrics and coding efficiency, it may not account for all aspects of video quality comprehensively. There might be cases where the algorithm optimizes for perceptual quality at the expense of other factors, such as compression efficiency or error resilience.

- **Dependency on Initial Population:** The genetic algorithm's performance can be sensitive to the quality of the initial population of candidate solutions. If the initial solutions are far from optimal, the algorithm may require more generations to converge to a satisfactory result.

- **Generalization to Diverse Content:** The algorithm's performance may vary depending on the content of the video. It could excel for certain types of videos while showing limitations for others. Achieving a more generalized and robust solution across a wide range of video content remains a challenge.

- **Lack of Real-world Testing:** The algorithm's evaluation is typically performed in controlled experimental settings. Real-world scenarios may introduce additional complexities, such as varying network conditions and device capabilities, which could affect its performance differently.

- **Scalability to Ultra High-Definition:** While the algorithm demonstrates scalability to different video resolutions, its applicability to ultra high-definition (UHD) or 8K content may require further optimization and testing.

- **Patent and Licensing Considerations:** Depending on the specific implementation details and the use of proprietary video coding tools, the algorithm may be subject to patent and licensing restrictions, potentially limiting its adoption.

– Continuous Evolution of Video Coding Standards: Video coding standards, such as H.266/VVC, continue to evolve. Keeping the algorithm up-to-date with the latest standards and ensuring compatibility with future revisions may require ongoing development efforts.

– Addressing these limitations and further optimizing the algorithm for various use cases will be essential for its successful adoption and practical application in the field of video coding.

**6. Conclusion.** In summary, this research introduces a novel approach using a genetic algorithm-driven method tailored for intra coding within the H.266/VVC framework. This approach effectively orchestrates the utilization of coding tools and MTT partitions, leading to a reduction in encoding time while preserving video quality. Empirical findings underscore the efficacy of this algorithm, highlighting its viability for real-world integration into H.266/VVC encoders. The proposed algorithm strikes a harmonious equilibrium between encoding efficiency and video quality, positioning it as a promising solution for expeditious intra coding within H.266/VVC.

## References

1. Fu T., Zhang H., Mu F., Chen H. Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding. IEEE International Conference on Multimedia and Expo (ICME). 2019. pp. 55–60. DOI: 10.1109/ICME.2019.00018.
2. Tsai Y.-H., Lu C.-R., Chen M.-J., Hsieh M.-C., Yang C.-M., Yeh C.-H. Visual Perception Based Intra Coding Algorithm for H.266/VVC. Electronics. 2023. vol. 12(9). no. 2079. DOI: 10.3390/electronics12092079.
3. Zhao J., Wu A., Zhang Q. SVM-Based Fast CU Partition Decision Algorithm for VVC Intra Coding. Electronics. 2022. vol. 11(14). no. 2147. DOI: 10.3390/electronics11142147.
4. Wang F., Wang Z., Zhang Q. FSVM- and DAG-SVM-Based Fast CU-Partitioning Algorithm for VVC Intra-Coding. Symmetry (Basel). 2023. vol. 15(5). no. 1078. DOI: 10.3390/sym15051078.
5. PyGAD: Genetic Algorithm in Python. GeneticAlgorithmPython, 2023. URL: <https://github.com/ahmedfgad/GeneticAlgorithmPython> (accessed 22.11.2023).
6. Zhang Q., Wang Y., Huang L., Jiang B. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. IEEE Access. 2020. vol. 8. pp. 117539–117550. DOI: 10.1109/ACCESS.2020.3004580.
7. Fan Y., Chen J., Sun H., Katto J., Jing M. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. IEEE Access. 2020. vol. 8. pp. 107900–107911. DOI: 10.1109/ACCESS.2020.3000565.
8. Yang H., Shen L., Dong X., Ding Q., An P., Jiang G. Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding. IEEE Trans. Circuits Syst. Video Technol. 2019. vol. 30. no. 6. pp. 1668–1682. DOI: 10.1109/TCSVT.2019.2904198.
9. Wu G., Huang Y., Zhu C., Song L., Zhang W. SVM Based Fast CU Partitioning Algorithm for VVC Intra Coding. IEEE International Symposium on Circuits and Systems (ISCAS). 2021. pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401614.

10. Zhang H., Yu L., Li T., Wang H. Fast GLCM-based Intra Block Partition for VVC. *Data Compression Conference (DCC)*. 2021. pp. 382–382. DOI: 10.1109/DCC50243.2021.00060.
11. Zhao J., Wu A., Jiang B., Zhang Q. ResNet-Based Fast CU Partition Decision Algorithm for VVC. *IEEE Access*. 2022. vol. 10. pp. 100337–100347. DOI: 10.1109/ACCESS.2022.3208135.
12. Chen M.-J., Lee C.A., Tsai Y.H., Yang C.M., Yeh C.H., Kau L.J., Chang C.Y. Efficient Partition Decision Based on Visual Perception and Machine Learning for H.266/Versatile Video Coding. *IEEE Access*. 2022. vol. 10. pp. 42141–42150. DOI: 10.1109/ACCESS.2022.3168155.
13. Huo J., Sun Y., Wang H., Wan S., Yang F., Li M. Unified Matrix Coding for NN Originated MIP in H.266/VVC. *ICASSP 2022 – 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022. pp. 1635–1639. DOI: 10.1109/ICASSP43922.2022.9746799.
14. Yang S., Hsiao S. H.266/VVC fast intra prediction using Sobel edge features. *Electron. Lett.* 2021. vol. 57. no. 1. pp. 11–13. DOI: 10.1049/el12.12011.
15. Chen C., Wang X. A Novel Selective Encryption Scheme for H.266/VVC Video. *IEEE International Conference on Image Processing (ICIP)*. 2023. pp. 1225–1229. DOI: 10.1109/ICIP49359.2023.10222774.
16. Kwon Y., Cho K. Design of High-performance Unified DST-7/DCT-8 Circuit for Video Encoding based on H.266/VVC. *Journal of the Institute of Electronic Engineers*. 2022. vol. 59. no. 1. pp. 10–18. DOI: 10.5573/ieie.2022.59.1.10.
17. Zhou Y.-G., Chen J.-J. Speed up H.266/VVC Intra Coding based on Statistical Heuristic and Deep Learning Methods. *International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan)*. 2023. pp. 451–452. DOI: 10.1109/ICCE-Taiwan58799.2023.10226774.
18. Regensky A., Herglotz C., Kaup A. Motion Plane Adaptive Motion Modeling for Spherical Video Coding in H.266/VVC. *IEEE International Conference on Image Processing (ICIP)*. 2023. pp. 875–879. DOI: 10.1109/ICIP49359.2023.10222661.
19. Gou A., Sun H., Liu C., Zeng X., Fan Y. A novel fast intra algorithm for VVC based on histogram of oriented gradient. *Journal of Visual Communication and Image Representation*. 2023. vol. 95. DOI: 10.1016/j.jvcir.2023.103888.
20. Liu L., Yang J. An Adaptive CU Split Method for VVC Intra Encoding. *International Journal of Information Technologies and Systems Approach (IJITSA)*. 2023. vol. 16. no. 2. pp. 1–17. DOI: 10.4018/IJITSA.322433.
21. Jing Z., Zhu W., Zhang Q. A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN. *Electronics*. 2023. vol. 12(9). no. 1963. DOI: 10.3390/electronics12091963.
22. Lee M., Song H., Park J., Jeon B., Kang J., Kim J.G., Lee Y.-L., Kang J.-W., Sim D. Overview of Versatile Video Coding (H.266/VVC) and Its Coding Performance Analysis. *IEIE Trans. Smart Process. Comput.* 2023. vol. 12. no. 2. pp. 122–154. DOI: 10.5573/IEIESPC.2023.12.2.122.
23. Park J., Lee J., Kim B., Jeon B. Learning-based Early Transform Skip Mode Decision for VVC Screen Content Coding. *IEEE Transactions on Circuits and Systems for Video Technology*. 2023. vol. 33. no. 10. pp. 6041–6056. DOI: 10.1109/TCSVT.2023.3257224.
24. Teng Q., Teng G., Li X., Ma R., An P., Yang Z. Lookahead Search-Based Low-Complexity Multi-Type Tree Pruning Method for Versatile Video Coding (VVC) Intra Coding. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. 2023. vol. E106.A. no. 3. DOI: 10.1587/transfun.2022EAP1022.

25. Wang Y., Dai P., Zhao J., Zhang Q. Fast CU Partition Decision Algorithm for VVC Intra Coding Using a MET-CNN. *Electronics*. 2022. vol. 11(19). no. 3090. DOI: 10.3390/electronics11193090.
26. Xu Q., Xu D., Wang H., Mi Z., Wang Z., Yan H. Detecting double H.266/VVC compression with the same coding parameters. *Neurocomputing*. 2022. vol. 514. pp. 231–244. DOI: 10.1016/j.neucom.2022.09.153.
27. Shang X., Li G., Zhao X., Zuo Y. Low complexity inter coding scheme for Versatile Video Coding (VVC). *Journal of Visual Communication and Image Representation*. 2023. vol. 90. DOI: 10.1016/j.jvcir.2022.103683.
28. CPH subset of the database for CU Partition of HEVC (CPH). 2018. URL: <https://github.com/HEVC-Projects/CPH> (accessed 22.11.2023).
29. UVG dataset. 2023. URL: <https://www.kaggle.com/datasets/minhngt02/uvg-yuv> (accessed 22.11.2023).

**Ibraheem Murooj Khalid Ibraheem** — Postgraduate student, Phystech school of radio engineering and computer technologies (FRKT), Department of multimedia technologies and telecommunications, Moscow Institute of Physics and Technology (MIPT); Assistant teacher, College of engineering, Department of computer engineering, Mustansiriyah University. Research interests: computer network, computer vision, video processing, machine learning, deep learning, software engineering. The number of publications — 7. [ibragim.m@phystech.edu](mailto:ibragim.m@phystech.edu); 9, Institutsky Lane, 141701, Dolgoprudny, Russia; office phone: +964(770)580-0699.

**Abdalameer Al-khafaji** — Postgraduate student, Institute of information technologies, Russian Technological University RTU MIREA. Research interests: information technology, deep learning, machine learning and software development. The number of publications — 8. [misnew6@gmail.com](mailto:misnew6@gmail.com); 78, Vernadskogo Av., 119454, Moscow, Russia; office phone: +7(985)177-9339.

**Hatif Naji Al-azzawi** — Postgraduate student, Phystech school of radio engineering and computer technologies (FRKT), Department of intelligent information systems and technologies, Moscow Institute of Physics and Technology (MIPT); Assistant teacher, College of engineering, department of computer engineering, University of Diyala. Research interests: image processing, artificial intelligence, computer vision, software engineering, deep learning, machine learning. The number of publications — 5. [al-azzawi.z@phystech.edu](mailto:al-azzawi.z@phystech.edu); 9, Institutsky Lane, 141701, Dolgoprudny, Russia; office phone: +964(774)030-9181.

**Acknowledgements.** This research is supported by the National Research University, Moscow Institute of Physics and Technology (MIPT), Phystech School of Radio Engineering and Computer Technologies (FRKT), Department of multimedia technologies and telecommunications, Moscow, Russia.

М.Х. ИБРАГИМ, А.И. АБДАЛАМИР, А.З. ХАТИФ НАДЖИ  
**ОСНОВАННЫЙ НА ГЕНЕТИЧЕСКОМ ПОДХОДЕ АЛГОРИТМ  
ВНУТРИКОДИРОВАНИЯ ДЛЯ H.266/VVC**

*Ибрагим И.Х., Абдаламир А.И., Хатиф Наджи А.З.* **Основанный на генетическом подходе алгоритм внутрикодирования для H.266/VVC.**

**Аннотация.** Представлен генетический подход для оптимизации внутреннего кодирования в H.266/VVC. Предлагаемый алгоритм эффективно выбирает инструменты кодирования и многотипные древовидные разбиения (МТТ) для достижения баланса между временем кодирования и качеством видео. Функция оценки пригодности, которая объединяет показатели восприятия и эффективности кодирования, используется для оценки качества каждого возможного решения. Результаты демонстрируют значительное сокращение времени кодирования без ущерба для качества видео. Предлагаемый алгоритм выбирает инструменты кодирования из набора доступных инструментов в H.266/VVC. Эти инструменты включают режимы внутреннего прогнозирования, единицы преобразования, параметры квантования и режимы энтропийного кодирования. Схема разбиения МТТ включает четыре типа разбиений: квадро-дерево, двоичное дерево, тройное дерево и квадро-двоичное дерево. Показатели восприятия используются для оценки визуального качества закодированного видео. Показатели эффективности кодирования используются для оценки эффективности кодирования закодированного видео. Функция оценки пригодности объединяет показатели восприятия и показатели эффективности кодирования для оценки качества каждого возможного решения.

**Ключевые слова:** генетический алгоритм, H.266/VVC, внутреннее кодирование, инструменты кодирования, разделы МТД, время кодирования, качество видео.

### Литература

1. Fu T., Zhang H., Mu F., Chen H. Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding. IEEE International Conference on Multimedia and Expo (ICME). 2019. pp. 55–60. DOI: 10.1109/ICME.2019.00018.
2. Tsai Y.-H., Lu C.-R., Chen M.-J., Hsieh M.-C., Yang C.-M., Yeh C.-H. Visual Perception Based Intra Coding Algorithm for H.266/VVC. Electronics. 2023. vol. 12(9). no. 2079. DOI: 10.3390/electronics12092079.
3. Zhao J., Wu A., Zhang Q. SVM-Based Fast CU Partition Decision Algorithm for VVC Intra Coding. Electronics. 2022. vol. 11(14). no. 2147. DOI: 10.3390/electronics11142147.
4. Wang F., Wang Z., Zhang Q. FSVM- and DAG-SVM-Based Fast CU-Partitioning Algorithm for VVC Intra-Coding. Symmetry (Basel). 2023. vol. 15(5). no. 1078. DOI: 10.3390/sym15051078.
5. PyGAD: Genetic Algorithm in Python. GeneticAlgorithmPython, 2023. URL: <https://github.com/ahmedfgad/GeneticAlgorithmPython> (accessed 22.11.2023).
6. Zhang Q., Wang Y., Huang L., Jiang B. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. IEEE Access. 2020. vol. 8. pp. 117539–117550. DOI: 10.1109/ACCESS.2020.3004580.
7. Fan Y., Chen J., Sun H., Katto J., Jing M. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. IEEE Access. 2020. vol. 8. pp. 107900–107911. DOI: 10.1109/ACCESS.2020.3000565.

8. Yang H., Shen L., Dong X., Ding Q., An P., Jiang G. Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* 2019. vol. 30. no. 6. pp. 1668–1682. DOI: 10.1109/TCSVT.2019.2904198.
9. Wu G., Huang Y., Zhu C., Song L., Zhang W. SVM Based Fast CU Partitioning Algorithm for VVC Intra Coding. *IEEE International Symposium on Circuits and Systems (ISCAS)*. 2021. pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401614.
10. Zhang H., Yu L., Li T., Wang H. Fast GLCM-based Intra Block Partition for VVC. *Data Compression Conference (DCC)*. 2021. pp. 382–382. DOI: 10.1109/DCC50243.2021.00060.
11. Zhao J., Wu A., Jiang B., Zhang Q. ResNet-Based Fast CU Partition Decision Algorithm for VVC. *IEEE Access*. 2022. vol. 10. pp. 100337–100347. DOI: 10.1109/ACCESS.2022.3208135.
12. Chen M.-J., Lee C.A., Tsai Y.H., Yang C.M., Yeh C.H., Kau L.J., Chang C.Y. Efficient Partition Decision Based on Visual Perception and Machine Learning for H.266/Versatile Video Coding. *IEEE Access*. 2022. vol. 10. pp. 42141–42150. DOI: 10.1109/ACCESS.2022.3168155.
13. Huo J., Sun Y., Wang H., Wan S., Yang F., Li M. Unified Matrix Coding for NN Originated MIP in H.266/VVC. *ICASSP 2022 – 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022. pp. 1635–1639. DOI: 10.1109/ICASSP43922.2022.9746799.
14. Yang S., Hsiao S. H.266/VVC fast intra prediction using Sobel edge features. *Electron. Lett.* 2021. vol. 57. no. 1. pp. 11–13. DOI: 10.1049/ell2.12011.
15. Chen C., Wang X. A Novel Selective Encryption Scheme for H.266/VVC Video. *IEEE International Conference on Image Processing (ICIP)*. 2023. pp. 1225–1229. DOI: 10.1109/ICIP49359.2023.10222774.
16. Kwon Y., Cho K. Design of High-performance Unified DST-7/DCT-8 Circuit for Video Encoding based on H.266/VVC. *Journal of the Institute of Electronic Engineers*. 2022. vol. 59. no. 1. pp. 10–18. DOI: 10.5573/ieie.2022.59.1.10.
17. Zhou Y.-G., Chen J.-J. Speed up H.266/VVC Intra Coding based on Statistical Heuristic and Deep Learning Methods. *International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan)*. 2023. pp. 451–452. DOI: 10.1109/ICCE-Taiwan58799.2023.10226774.
18. Regensky A., Herglotz C., Kaup A. Motion Plane Adaptive Motion Modeling for Spherical Video Coding in H.266/VVC. *IEEE International Conference on Image Processing (ICIP)*. 2023. pp. 875–879. DOI: 10.1109/ICIP49359.2023.10222661.
19. Gou A., Sun H., Liu C., Zeng X., Fan Y. A novel fast intra algorithm for VVC based on histogram of oriented gradient. *Journal of Visual Communication and Image Representation*. 2023. vol. 95. DOI: 10.1016/j.jvcir.2023.103888.
20. Liu L., Yang J. An Adaptive CU Split Method for VVC Intra Encoding. *International Journal of Information Technologies and Systems Approach (IJITSA)*. 2023. vol. 16. no. 2. pp. 1–17. DOI: 10.4018/IJITSA.322433.
21. Jing Z., Zhu W., Zhang Q. A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN. *Electronics*. 2023. vol. 12(9). no. 1963. DOI: 10.3390/electronics12091963.
22. Lee M., Song H., Park J., Jeon B., Kang J., Kim J.G., Lee Y.-L., Kang J.-W., Sim D. Overview of Versatile Video Coding (H.266/VVC) and Its Coding Performance Analysis. *IEIE Trans. Smart Process. Comput.* 2023. vol. 12. no. 2. pp. 122–154. DOI: 10.5573/IEIESPC.2023.12.2.122.
23. Park J., Lee J., Kim B., Jeon B. Learning-based Early Transform Skip Mode Decision for VVC Screen Content Coding. *IEEE Transactions on Circuits and Systems for*



- Video Technology. 2023. vol. 33. no. 10. pp. 6041–6056. DOI: 10.1109/TCSVT.2023.3257224.
24. Teng Q., Teng G., Li X., Ma R., An P., Yang Z. Lookahead Search-Based Low-Complexity Multi-Type Tree Pruning Method for Versatile Video Coding (VVC) Intra Coding. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2023. vol. E106.A. no. 3. DOI: 10.1587/transfun.2022EAP1022.
  25. Wang Y., Dai P., Zhao J., Zhang Q. Fast CU Partition Decision Algorithm for VVC Intra Coding Using an MET-CNN. Electronics. 2022. vol. 11(19). no. 3090. DOI: 10.3390/electronics11193090.
  26. Xu Q., Xu D., Wang H., Mi Z., Wang Z., Yan H. Detecting double H.266/VVC compression with the same coding parameters. Neurocomputing. 2022. vol. 514. pp. 231–244. DOI: 10.1016/j.neucom.2022.09.153.
  27. Shang X., Li G., Zhao X., Zuo Y. Low complexity inter coding scheme for Versatile Video Coding (VVC). Journal of Visual Communication and Image Representation. 2023. vol. 90. DOI: 10.1016/j.jvcir.2022.103683.
  28. CPH subset of the database for CU Partition of HEVC (CPH). 2018. URL: <https://github.com/HEVC-Projects/CPH> (accessed 22.11.2023).
  29. UVG dataset. 2023. URL: <https://www.kaggle.com/datasets/minhngt02/uvg-yuv> (accessed 22.11.2023).

**Ибрагим Мурудж Халид Ибрагим** — аспирант, физико-техническая школа радиотехники и компьютерных технологий (FRKT), кафедра мультимедийных технологий и телекоммуникаций, Московский физико-технический институт (МФТИ); ассистент преподавателя, инженерный колледж, факультет компьютерной инженерии, Университет Мустансирия. Область научных интересов: компьютерные сети, компьютерное зрение, обработка видео, машинное обучение, глубокое обучение, разработка программного обеспечения. Число научных публикаций — 7. [ibragim.m@phystech.edu](mailto:ibragim.m@phystech.edu); Институтский переулок, 9, 141701, Долгопрудный, Россия; р.т.: +964(770)580-0699.

**Абдаламир Аль-Хафаджи Исраа М.** — аспирант, институт информационных технологий, Российский технологический университет МИРЭА. Область научных интересов: информационные технологии, глубокое обучение, машинное обучение и разработка программного обеспечения. Число научных публикаций — 8. [misnew6@gmail.com](mailto:misnew6@gmail.com); проспект Вернадского, 78, 119454, Москва, Россия; р.т.: +7(985)177-9339.

**Хатиф Наджи Аль-Аззави Зобед** — аспирант, физико-техническая школа радиотехники и компьютерных технологий (FRKT), кафедра интеллектуальных информационных систем и технологий, Московский физико-технический институт (МФТИ); ассистент преподавателя, инженерный колледж, факультет компьютерной инженерии, Университет Дияла. Область научных интересов: обработка изображений, искусственный интеллект, компьютерное зрение, разработка программного обеспечения, глубокое обучение, машинное обучение. Число научных публикаций — 5. [al-azzavi.z@phystech.edu](mailto:al-azzavi.z@phystech.edu); Институтский переулок, 9, 141701, Долгопрудный, Россия; р.т.: +964(774)030-9181.

**Поддержка исследований.** Данное исследование проводится при поддержке Национального исследовательского университета, Московского физико-технического института (МФТИ), Физико-технической школы радиотехники и компьютерных технологий (FRKT), факультета мультимедийных технологий и телекоммуникаций, Москва, Россия.

Б.Г. МАЙОРОВ  
**МОДЕЛИ СОСТАВНЫХ ГАРМОНИЧЕСКИХ ПОЛУВОЛН И  
СВЯЗЬ ДИСКРЕТИЗАЦИИ ВРЕМЕНИ С ЭНТРОПИЕЙ  
ВРЕМЕННЫХ ПАРАМЕТРОВ СИГНАЛОВ**

*Майоров Б.Г. Модели составных гармонических полувольт и связь дискретизации времени с энтропией временных параметров сигналов.*

**Аннотация.** Целью данного исследования является определение связи энтропии временных параметров сигналов в робастной системе управления с величиной дискретизации системного времени (в развитие работ trspy 1185, trspy 1274). В качестве примера объекта исследования рассмотрен процесс и его сигналы экстренного торможения высокоскоростного состава при наличии скольжения колёс по рельсам. Решена задача нахождения абсолютной погрешности ступенчатой и линейной интерполяции сигнала управления по равномерным выборкам из него с применением моделей составных гармонических полувольт. Предварительно, при обследовании объекта управления, определяются максимальные величины параметров сигнала и полувольт: скорость, ускорение и резкость. Параметры спектра отсутствуют по причине большой инерционности объектов управления, процессов и сигналов. Для определения величин интервалов равномерной дискретизации времени рассмотрены две группы моделей «гармонических полувольт». Первая группа моделей описывается гармоническими функциями времени, параметры которых согласованы. Вторая группа моделей описывается составными гармоническими функциями времени, тем самым согласуются временные параметры сигналов. Доказано, что при увеличении энтропии максимальных величин параметров сигналов увеличивается величина интервала дискретизации времени без увеличения погрешности интерполяции. Таким образом, величина энтропии параметров сигналов служит индикатором их рассогласованности. Приведены результаты моделирования и графики, полученные в среде математического пакета MathCAD. Результаты предназначены для оптимизации загрузки задачами ввода и первичной обработки информации процессоров в робастных системах автоматизации реального времени, например, используемых для управления высокоскоростными поездами при штатном экстренном торможении и экстренном торможении в условиях скольжения или юза.

**Ключевые слова:** робастные системы, реальное время, дискретизация времени, интерполяция нулевого и первого порядка, составные, гармонические функции, энтропия параметров сигналов, модель торможения с юзом высокоскоростных поездов.

**1. Введение.** Областью применения результатов настоящей статьи является использование полученных моделей при проектировании систем реального времени (СРВ) для объектов, управляемых, преимущественно, в автоматическом режиме. В подобных технических системах для первичной обработки аналоговых сигналов [1, 2] применяют бортовые процессоры. При этом часто отсутствует (неизмерима) информация о частотном спектре сигналов управляемых процессов в большинстве случаев в связи с большой инерционностью объектов управления. Эти сигналы являются сигналами объекта управления и их можно назвать

сигналами с трудно определяемым спектром [1, 2, 3]. Для наглядности, в дальнейшем, получим модель для описания интервала экстренного торможения высокоскоростного состава с юзом (скольжением) колёсных пар, сигналы процесса которого не имеет спектра. Правила экстренного торможения реализуют в соответствии с Европейской системой управления поездами (European Train Control System (ETCS)), где они связаны с регулированием ускорения (замедления) в зависимости от способности торможения поезда, данных рельефа и меняющейся погоды на маршруте движения [4 – 8].

Для первичной обработки сигналов с трудно определяемым спектром в системах реального времени необходимо:

- определять величину постоянного шага равномерной дискретизации сигнала процесса по времени по результатам замеров максимальных величин амплитудно-временных параметров сигнала на этапе предварительного обследования объекта управления: допустимого отклонения –  $\Delta$  от линейной аппроксимации соседних выборок, максимальной скорости –  $V$ , максимального ускорения –  $a$  и максимальной резкости движения –  $R$ ;

- обработку информации и выдачу сигнала управления на исполнительные органы СРВ проводить за один или два интервала дискретизации времени  $T_g$ , где индекс «г» – от слова гармонический;

- восстанавливать сигнал между выборками при помощи интерполяции нулевого (ступени), первого порядка (прямые) [1], при этом сохранять значения сигнала без изменений в точках выборок [9 – 11].

Критерием оптимальной обработки таких сигналов в СРВ является максимум интервала дискретизации времени  $T_g$ . Очевидным является тот факт, что чем больше  $T_g$ , тем меньше выборок будет сделано из одного сигнала процесса с сохранением заданной величины погрешности интерполяции  $\Delta$ , тем больше сигналов процессов можно подвергнуть первичной обработке в единицу времени в СРВ [12].

Задача и модель для определения величины интервала равномерной дискретизации времени  $T_g$  сигнала должна математически связывать максимальные амплитудно-временные параметры сигнала  $\{\Delta, V, a, R\}$  и интерполяцию нулевого или первого порядка по равномерным выборкам.

Наличие сигналов с трудно определяемым спектром на объекте управления свидетельствуют о неопределённости информационного описания объекта управления СРВ. Такие СРВ можно отнести к робастным системам управления [11, 12]. Модели робастных процессов и сигналов традиционно описывают в амплитудно-временной области. В системах управления роботом траекторию

оптимального процесса управления задают в виде сопрягаемых парабол [9, 13, 14]. Моделируют равноускоренное движение манипулятора руки квадратичным трёхчленом [15], в качестве коэффициентов трёхчлена используют максимальные значения ускорения и скорости сигнала. Осуществляют сопряжение параболами смежных участков траектории движения космического аппарата [16]. В оптико-электронных системах для обнаружения и сопровождения подвижных объектов, которые имеют значительные угловые скорости и ускорения, используют модель «машина Дубинса» [17 – 19]. Представляют сейсмические события в виде линейных степенных функций для упрощения анализа больших объёмов данных [20].

Известны две формулы В.Н. Хлистунова [21] в виде линейного и квадратичного остаточного члена формулы Лагранжа в измерительной технике и в других применениях систем управления [2]. Обобщением этих двух формул является формула [22], которая обосновывает существенное увеличение интервала дискретизации времени при сохранении точности интерполяции сигнала. Положительный результат дают модели в виде гармонических полуоволн [9, 23, 25]. Во всех примерах, в качестве основного физико-математического обоснования моделей, применены основополагающие идеи *принципа максимума Л.С. Понтрягина и понятия частоты Найквиста (Nyquist H.)* [2, 26].

**Постановка задачи:** Определение максимального интервала равномерной дискретизации времени  $\max T_r = 1/2f$  при неизвестной частоте Найквиста  $2f$  (где неизвестна  $f$  частота среза спектра), когда задана погрешность линейной интерполяции соседних выборок  $\max p^{(0)}(t) = \Delta$ , найдены и замерены, предварительно, максимальные производные сигнала процесса, максимальные параметры  $\{\Delta, V, a, R\}$ . Измерения максимальных параметров проводятся не в реальном времени, а на предварительном этапе обследования объекта управления, адаптирующая интерполяция в этом случае не применяется. Для измерений используют, как правило, цифровые аналитические осциллографы или анализаторы сигналов [3].

В упомянутых выше исследованиях не отражено: как изменение величины рассогласования (энтропии  $H$ ) максимальных величин амплитудно-временных параметров процесса (сигнала)  $\{\Delta, V, a, R\}$  связано с величиной интервала дискретизации времени или интервалом выдачи управляющих воздействий на объект управления?

В тоже время энтропия является математическим индикатором и характеристикой степени беспорядка или рассогласования, поэтому необходимо исследовать её роль в применении к соотношению

величин  $\{\Delta, V, a, R\}$ . Это позволит определять (индексировать) в произвольный момент времени необходимость применения результатов исследования предложенных моделей сигналов с трудно определяемым спектром к управлению в робастных системах реального времени.

В данной работе применена методика построения и исследованы свойства гармонических полуволн с согласуемыми параметрами для моделирования сигналов в робастных системах. Это позволяет создать модели гармонических полуволн с согласуемыми параметрами и определить связь энтропии рассогласования с величиной интервалов дискретизации сигналов без потери точности интерполяции процесса (сигнала) по выборкам.

**2. Свойства моделей гармонических полуволн сигнала управляемого процесса и основные выражения.** Для устранения упомянутых особенностей и нахождения зависимости интервала дискретизации времени от коэффициентов рассогласования максимальных скорости и ускорения сигналов или от степени энтропии этих параметров применим допущение и рассмотрим две группы моделей: с согласованными амплитудно-временными параметрами и с согласуемыми параметрами.

*Определение 1. «Согласованная гармоническая полуволна является выпуклой гармонической траекторией (функцией времени, заданной одной формулой), по которой сигнал процесса изменяется между значениями в соседних выборках при одновременном учёте максимальных параметров  $\{\Delta, V, a, R\} (1 \leq n \leq 3)$  на временном интервале  $0 \leq t \leq T_r$ . При этом  $(n+1)$  максимальный параметр входит в равенство, названное условием согласованности параметров».*

Допущение (гипотеза подобия). Гармоническая полуволна имеет максимальные временные параметры  $\{\max |r^{(i)}(t)|, \text{ где } 1 \leq i \leq n\}$ , которые имеет сигнал  $r(t)$ , поэтому может быть использована, как математическая модель для исследования взаимозависимостей параметров сигнала  $r(t)$  и величины шага равномерной дискретизации  $T_r$  сигнала  $r(t)$  по времени [9, 25].

Общие свойства гармонической полуволны. Модель гармонической полуволны (ГПВ<sup>r</sup>) является выпуклой функцией времени, симметрична относительно прямой  $t = 0.5T_r$ , начало и конец полуволны совпадают с двумя соседними выборками или замыкаются интервалом равномерной дискретизации  $T_r$  сигнала  $r(t)$  процесса, амплитуда полуволны равна  $\max^{(0)}(0.5T_r) = \Delta$ .

Допущению и Общим свойствам соответствует математическая модель в виде простейшей гармонической функции времени [9, 25]:

**3. Согласованная модель гармонической полуволны и нахождение шага дискретизации времени сигнала процесса.** Уравнение простейшей гармонической полуволны [9, 25]:

$$p(t) = \Delta \sin \omega t, \quad (1)$$

где  $0 \leq t \leq T_r$ ,  $T_r = T/2 = 1/2f$  – интервал Найквиста,  $f$  – частота среза в спектре сигнала  $p(t)$  после цифрового фильтра нижних частот,  $T$  – период частоты среза,  $\omega = 2\pi f$  – круговая частота,  $\Delta$  – амплитуда гармонической полуволны или максимальная погрешность линейной аппроксимации двух выборок из полуволны в моменты времени  $t=0$  и  $t=T_r$ . Первая производная ( $i=1$ ) от гармонической полуволны:

$$p^{(1)}(t) = \omega \Delta \cos \omega t = V(t), \quad (2)$$

где ( $i=1$ ) – порядок производной,  $1 \leq i \leq n$ . Вторая ( $i=2$ ) и третья ( $i=3$ ) производные от гармонической полуволны [25]:

$$p^{(2)}(t) = -\omega^2 \Delta \sin \omega t = a(t); \quad p^{(3)}(t) = -\omega^3 \Delta \cos \omega t = R(t). \quad (3)$$

Гармоническая полуволна и три производные от неё изображены на рисунке 1.

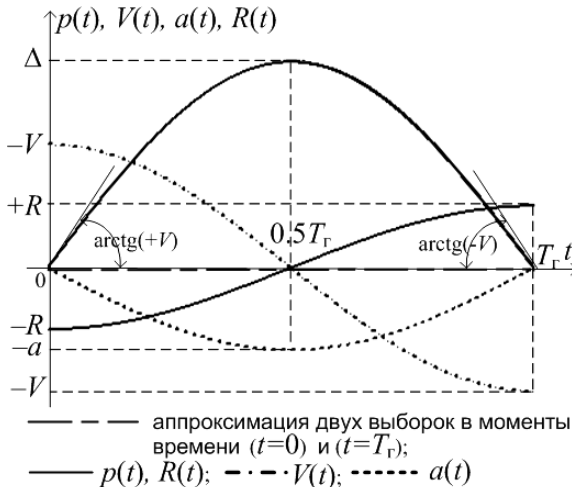


Рис. 1. Согласованная, гармоническая полуволна и три производных

Выражение (1) или гармоническая полуволна (ГПВ<sup>n</sup>) описывает математически половину  $T_r = T/2 = 1/2f$  периода  $T$  частоты среза  $f$  в спектре сигнала  $p(t)$ . Частота  $2f$  широко используется в теории связи, управления и в информатике [4 – 6].

Первая выборка из сигнала  $p(t)$  осуществлена в момент времени  $t = 0$ , вторая – в момент времени  $t = T_r = T/2$ . Линейная интерполяция между двумя выборками изображена двух штриховой линией. Погрешность такой интерполяции  $\Delta$  достигает максимального значения в точке  $(0.5T_r, \Delta)$ , которая является вершиной гармонической полуволны. Проходящая через вершину вертикальная линия  $t = 0.5T_r = 0.25T$  служит её осью симметрии. Связь круговой частоты  $\omega$  и интервала гармонической полуволны  $T_r$  запишем в виде выражения [25]:

$$\omega = 2\pi f = \frac{2\pi}{T} = \frac{\pi}{T_r}. \quad (4)$$

Определим интервал гармонической полуволны  $T_r$  через максимальные параметры сигнала, которые должны быть измерены при исследовании спектра сигнала  $p(t)$  или получены при обработке результатов обследования объекта управления, если сигнал процесса является сигналом с трудно определяемым спектром. В выражение первой производной (2) подставим моменты времени выборок из сигнала  $p(t)$ :  $p(t = 0) = 0$  и  $p(t = T_r) = 0$  и приравняем двум её максимальным значениям:  $\max p^{(1)}(t = 0, t = T_r) = |\pm V|$ , учтём (4), получим [25]:

$$T_r = \pi \frac{\Delta}{V}. \quad (5)$$

В выражение второй производной (3) подставим момент времени  $(t = 0.5T_r)$ , при котором наступает максимум ускорения сигнала  $p(t)$  и приравняем её максимальному значению:  $\max p^{(2)}(t = 0.5T_r) = |-a|$ , аналогично при известном значении  $\max p^{(3)}(t = 0, t = T_r) = |\pm R|$ . Используем (4), получим цепь равных элементов:

$$T_r = \pi \frac{\Delta}{V} = \pi \frac{V}{a} = \pi \left(\frac{\Delta}{a}\right)^{1/2} = \pi \frac{a}{R} = \pi \left(\frac{\Delta}{R}\right)^{1/3}, \quad (6)$$

где очевидны два частных условия согласования параметров  $\{\Delta, V, a, R\}$  модели:  $V^2 = \Delta a$ ,  $a^2 = VR$ . Общее условие согласования:  $Va = \Delta R$  получим по схеме перемножения правых и левых частей:

$$\left. \begin{aligned} & \times V^2 = \Delta a \\ & a^2 = VR \end{aligned} \right\} \rightarrow Va = \Delta R. \quad (7)$$

С учётом (4) преобразуем (6) в круговую частоту  $\omega$ :

$$\omega = \frac{V}{\Delta} = \frac{a}{V} = \left(\frac{a}{\Delta}\right)^{1/2} = \left(\frac{R}{V}\right)^{1/2} = \frac{R}{a} = \left(\frac{R}{\Delta}\right)^{1/3}. \quad (8)$$

**4. Согласование максимальных параметров  $\{\Delta, V_A, a\}$  сигнала процесса отрезками прямых линий.** Рисунок 2 позволяет дать определение и записать математическую модель, которая показывает возможность согласования максимальных параметров  $\{\Delta, V_A, a\}$  сигнала  $p(t)$  [9]:

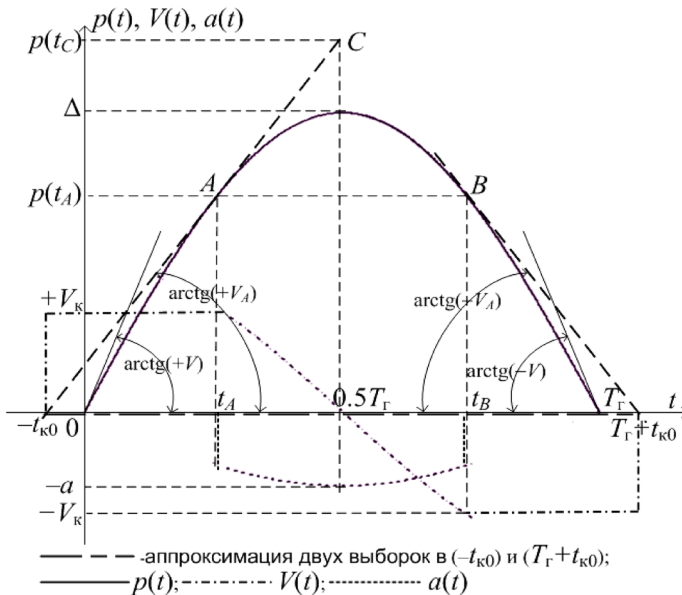


Рис. 2. Составная полуволна, согласуемая двумя отрезками прямых линий,  $\pm V_A = \pm \text{const} \leq V$

Графическая модель траектории составной полуволны характеризуется максимальной скоростью  $V_A$ , которая постоянна в начале и конце траектории  $\pm V_A = \pm \text{const}$ . Поэтому эту полуволну



назовём «гармонической полувольтной, согласуемой двумя отрезками прямых линий» и введём индекс «гл» в обозначение гармонической полувольтны ( $\text{ГПВ}^{\text{гл}}$ ) и интервала времени действия составной полувольтны  $T_{\text{гл}}$ . Первая выборка из составной согласуемой полувольтны осуществлена в момент времени  $(-t_{\text{к0}})$ , вторая – в момент времени  $(T_{\text{г}} + t_{\text{к0}})$  (рисунок 2). Таким образом, из рисунка 2 следует величина интервала продолжительности составной полувольтны:

$$T_{\text{гл}} = T_{\text{г}} + 2|t_{\text{к0}}|. \quad (9)$$

Между двумя выборками сигнал отклоняется по составной полувольтне, состоящей из трёх частей:

- на интервале времени  $(-t_{\text{к0}} \leq t \leq t_A)$  с постоянной скоростью  $V_A \leq (\Delta a)^{1/2}$ , по прямой линии касательной к гармонической полувольтне синусоиды (1) в точке  $A$ ;
- между точками касания  $A$  и  $B$  линий касательных к синусоиде (1) по верхней части гармонической полувольтны (1);
- на интервале времени  $t_B \leq t \leq (T_{\text{г}} + t_{\text{к0}})$  с постоянной скоростью  $-V_A \geq -(\Delta a)^{1/2}$ , по прямой линии касательной к (1) в точке  $B$ .

Для нахождения интервала полувольтны  $T_{\text{гл}}$  по выражению (9) определим момент времени  $t_{\text{к0}}$ , для чего найдём касательную к полувольтне синусоиды (1) в точке  $A$ . Первую производную (2) от (1) в точке  $A$  приравняем скорости  $V_A$ , введём коэффициент согласования скоростей  $0 \leq k \leq 1$ , запишем  $V_A = kV$ , момент времени  $t_A$  подставим в (1), получим промежуточный результат в виде координат точки  $A$ :

$$t_A = \frac{1}{\omega} \arccos(k) \quad \text{и} \quad p(t_A) = \Delta(1 - k^2)^{1/2}. \quad (10)$$

Запишем уравнение касательной прямой через точку  $A[t_A, p(t_A)]$  под углом  $\arctg V_A$ :  $p(t) - p(t_A) = V_A(t - t_A)$ . Пересечение касательной с осью абсцисс ( $0t$ ) при  $t = t_{\text{к0}}$ ,  $p(t_{\text{к0}}) = 0$  (рисунок 2) даёт:

$$t_{\text{к0}} = \frac{1}{\omega} \left[ \left( \frac{1}{k^2} - 1 \right)^{1/2} - \arccos(k) \right].$$

Используем (6), (8), получим  $\omega = \pi/T_{\text{г}}$ , преобразуем и подставим  $|t_{\text{к0}}|$  в (9), введем относительную величину интервала составной полувольтны согласуемой отрезками прямых  $\tau_{\text{гл}} = T_{\text{гл}} / T_{\text{г}}$ , получим:

$$\tau_{\text{гл}} = 1 + \frac{2}{\pi} \left[ \left( \frac{1}{k^2} - 1 \right)^{1/2} - \arccos(k) \right], \quad (11)$$

где  $0 \leq k \leq 1$ ; частное условие согласования параметров  $\{\Delta, V_A, a\}$  процесса:  $V_A = k(\Delta a)^{1/2}$ . Для наглядности сравнения величин интервалов времени согласованной гармонической полувольты  $T_{\text{г}}$  (6) и согласуемой линией составной полувольты  $T_{\text{гл}}$  (11) введем относительную величины интервала гармонической полувольты  $\tau_{\text{г}} = T_{\text{г}}/T_{\text{г}} = 1$ . Получим графики относительных величин  $\tau_{\text{гл}}$  и  $\tau_{\text{г}}$  (рисунок 3) [25].

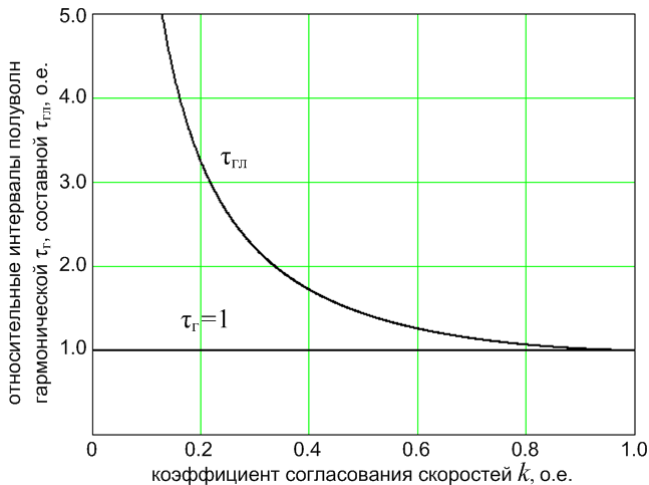


Рис. 3. Зависимости относительных интервалов согласованной гармонической и согласуемой составной полувольт от коэффициента  $k$

Графики показывают, что при уменьшении коэффициента согласования  $k$  или уменьшении максимальной скорости изменения сигнала  $V_A = kV$  происходит увеличение относительного интервала времени составной полувольты  $\tau_{\text{гл}} \geq 1$ . Этот результат позволяет сформулировать новое информационное свойство сигналов процессов:

*Свойство 1: Уменьшение величины коэффициента согласования  $k$  параметров  $\max\{\Delta, V_A, a\}$  сигнала  $p(t)$  приводит к увеличению интервала времени  $T_{\text{гл}}$  составной полувольты (или  $\tau_{\text{гл}} \geq 1$ ). При этом максимальная абсолютная погрешность измерения равна амплитуде согласованной гармонической полувольты  $\Delta$  или не изменяется [25].*

Из вышесказанного следует определение согласуемой отрезками прямых линий составной гармонической полуволны (ГПВ<sup>гп</sup>).

**Определение 2.** «Составная ГПВ<sup>гп</sup> является выпуклой гармонической функцией модуля времени, заданной последовательно несколькими формулами, которые описывают изменение сигнала процесса между соседними выборками на временном интервале  $0 \leq t \leq T_{\text{гл}}$ . При этом три максимальных параметра  $\{\Delta, V_A, a\}$  входят в условие согласованности параметров (УСП):  $V_A < (\Delta a)^{1/2}$  или  $V_A = k (\Delta a)^{1/2}$ , где  $k = V_A/V \leq 1$  – коэффициент согласования скоростей».

**5. Согласование максимальных параметров  $\{\Delta, V_A, a\}$  сигнала процесса частями гармонических функций.** На рисунке 4 приведено изображение составной полуволны согласованной двумя гармоническими линиями (ГПВ<sup>2г</sup>). Траектория ГПВ<sup>2г</sup> задаётся составными функциями, заданными тремя формулами гармонических функций.

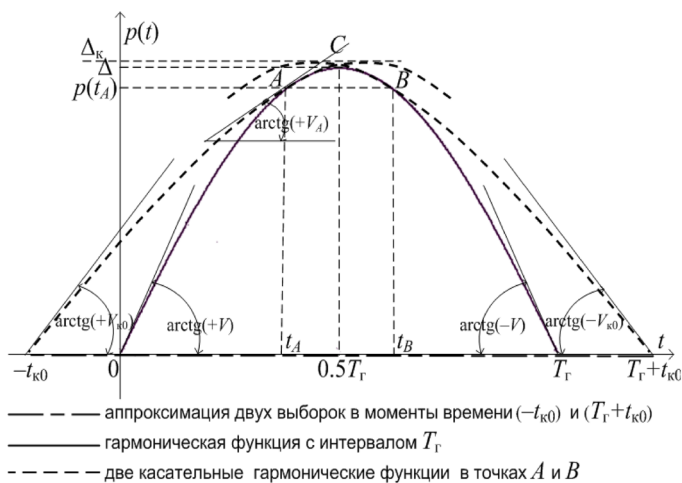


Рис. 4. Составная полуволна ГПВ<sup>2г</sup>, согласуемая двумя частями гармонических полуволн при  $V_A \leq V_{k0} \leq V$

Рисунок 4 поясняет связь параметров  $\{\Delta, V_A, a\}$  сигнала  $p(t)$ , позволяет записать математическую модель ГПВ<sup>2г</sup>, которая согласует эти параметры. Участки согласующей полуволны описываются двумя симметричными частями гармонической функции, где нижний индекс «к» обозначает внешнее касание левой части гармонической функции в точке A к гармонической полуволне (1) [9]:

$$p(t) = \Delta_k \sin(\omega_k t + \varphi_0), \quad (12)$$

где  $\Delta_k$ ,  $\varphi_0$ ,  $\omega_k$  – соответственно, амплитуда, фаза и круговая частота согласующей полуволны;  $(-t_{k0}) \leq t \leq (T_\Gamma + t_{k0})$ ;  $t_{k0} = -\varphi_0/\omega_k$ ;  $\omega_k = 2\pi/T_k$ ;  $\omega_k = V_k/\Delta_k$  по аналогии с (8). Так как параметры  $\{\Delta, V_A, a\}$  полуволны на рисунке 4 согласуются двумя одинаковыми гармоническими полуволнами, введём индекс «2Г» в обозначение интервала времени полуволны  $T_{2Г}$ .

Предположим, что для сигнала  $p(t)$  процесса предварительно замерили параметры  $\{\Delta, V_A, a\}$ . Причём величина максимальной скорости в точке  $A$  равна  $V_A = kV = k(\Delta a)^{1/2}$ , где  $0 \leq k \leq 1$ . Процесс отклонения полуволны от прямой, аппроксимирующей две соседние выборки в моменты времени  $(-t_{k0})$  и  $(T_\Gamma + t_{k0})$ , происходит по трём участкам:

- в интервале времени  $(-t_{k0} \leq t \leq t_A)$ , по левой части гармоники (12) от точки  $(-t_{k0})$  до точки  $A$ , при этом скорость  $V_A \leq V_{k0} \leq V = (\Delta a)^{1/2}$ ;

- между точкой касания  $A$  гармонической функции (12) к синусоиде (1) в момент времени  $t = t_A$  до точки её пересечения с осью симметрии в момент времени  $t = 0.5T_\Gamma$ ;

- в интервале времени  $(0.5T_\Gamma \leq t \leq (T_\Gamma + t_{k0}))$  симметрично второму и первому участкам относительно оси симметрии  $t = 0.5T_\Gamma$ .

Интервал времени составной полуволны  $T_{2Г}$  можно определить по выражению (12) при  $p(-t_{k0}) = 0$ , с учётом величины фазы гармонической траектории наибольшего отклонения  $|t_{k0}| = \varphi_0/\omega_k$  и  $\omega_k = 2\pi/T_k$ :

$$T_{2Г} = T_\Gamma + \frac{1}{\pi} T_k \varphi_0 = T_\Gamma \left( 1 + \frac{2}{\pi} \varphi_0 r \right), \quad (13)$$

где  $r = T_k/2T_\Gamma$ ,  $T_k \geq (T = 2T_\Gamma)$ ,  $r \geq 1$  [9]. Для определения фазы  $\varphi_0$  воспользуемся равенством скоростей  $V_A$  в точке взаимного касания  $A(t_A, p(t_A))$  гармонических функций (1) и (12). Производные этих функций по времени преобразуем  $t_A = (1/\omega) \arccos(k) = (1/\omega_k)(\arccos(m) - \varphi_0)$ . Используем условия согласования параметров процесса:  $V_A = kV$ ,  $V_A = mV_{k0}$ . Найдём отношение выражения (4)  $\omega = \pi/T_\Gamma$  и ему подобного выражения  $\omega_k = 2\pi/T_k$ , которое равно  $1/r = \omega_k/\omega = 2T_\Gamma/T_k$ . Получим:

$$\varphi_0 = \arccos(m) - \frac{1}{r} \arccos(k). \quad (14)$$

Найдём связь между величинами коэффициентов  $k$ ,  $m$  согласования параметров сигнала и коэффициентом  $r$  [9].

Выражение (1) при  $t = t_A$  запишем в виде  $\Delta \sin \omega t_A = p(t_A)$  и последовательно преобразуем его по известной тригонометрической формуле  $\arcsin x = \arccos[(1 - x^2)^{1/2}]$ , получим:

$$\omega t_A = \arcsin [p(t_A)/\Delta] = \arccos [1 - (p(t_A)/\Delta)^2]^{1/2}. \quad (15)$$

Первая производная от (1) при  $t = t_A$  равна скорости  $V(t_A) = V_A$ , из (2) запишем:  $\Delta \omega \cos \omega t_A = V_A$ . В это выражение подставим вторую часть выражения (15) и применим тригонометрическую формулу  $\cos \arccos x = x$ . Получим выражение  $p^2(t_A) = \Delta^2 - (V_A/\omega)^2$ , в которое последовательно подставим  $\omega = V/\Delta$  из (8) и  $V_A = kV$ , в итоге получим:  $p^2(t_A) = \Delta^2(1 - k^2)$ . Аналогично преобразуем выражение (12), при этом используем  $\omega_k = V_{k0}/\Delta_k$  аналогичное (8), и  $V_A = mV_{k0}$ . Получим выражение:  $p^2(t_A) = \Delta_k^2(1 - m^2)$ . Два выражения для нахождения  $p^2(t_A)$  преобразуем совместно через их отношение, получим:

$$\frac{\Delta^2}{\Delta_k^2} = \frac{1 - m^2}{1 - k^2}. \quad (16)$$

Определим отношение амплитуд составной траектории отклонения  $\Delta$  и  $\Delta_k$  через коэффициенты согласования параметров сигнала  $m$ ,  $k$  и коэффициент  $r$ . Для этого последовательно применим выражения (8)  $\Delta = V/\omega$ ,  $\Delta_k = V_{k0}/\omega_k$ , выражения (4)  $\omega = \pi/T_r$ ,  $\omega_k = 2\pi/T_k$ , выражения  $V_A = kV$ ,  $V_A = mV_{k0}$ , получим частное  $\Delta/\Delta_k = m/kr$ . Подставим это выражение в (16) и выделим коэффициент согласования  $r \geq 1$ , в соответствии с (13):

$$r = \frac{m}{k} \left( \frac{1 - k^2}{1 - m^2} \right)^{1/2} \geq 1. \quad (17)$$

Выражение (17) умножим на (14) ( $\varphi_0 \cdot r$ ) и результат подставим в (13). Введем относительную величину интервала гармонической составной полуволны  $\tau_{2r} = T_{2r}/T_r$ . В результате преобразований получим искомое выражение:

$$\tau_{2r} = \frac{T_{2r}}{T_r} = 1 + \frac{2}{\pi} \left[ \frac{m}{k} \left( \frac{1 - k^2}{1 - m^2} \right)^{1/2} \cdot \arccos(m) - \arccos(k) \right], \quad (18)$$

где  $0 \leq k \leq m \leq 1$ . Графики  $\tau_{2r}$  для  $m = 0.2$ ,  $m = 0.5$ ,  $m = 0.8$  для наглядности совместим с графиками  $\tau_{r1}(k)$  и  $\tau_r(k)$  (рисунки 3 и 5).

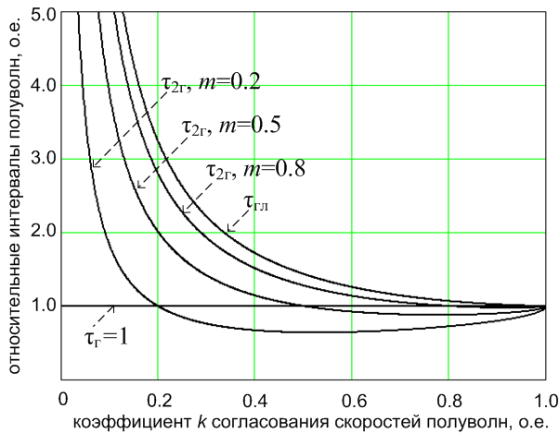


Рис. 5. Примеры зависимостей относительных интервалов гармонических полуволн от коэффициентов согласования скоростей  $k, m$

Сравним графики на рисунке 5 с графиками величин энтропии  $H_{2r}\{\Delta, V_A, a\} = \log_2(\tau_{2r})$  (рисунок 6) для тех же величин  $m$  пяти примеров согласований параметров. Каждый график  $\tau_{2r}$  определён в своей верхней части ( $\tau_{2r} \geq \tau_r=1$ ), при коэффициенте согласования из замкнутого интервала  $0 \leq k \leq m$ , что следует из (17) и (13) [9].

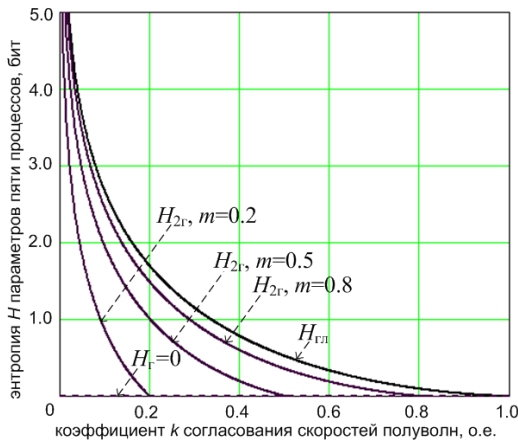


Рис. 6. Примеры зависимостей энтропии параметров составных полуволн от коэффициентов согласования скоростей  $k, m$

Правые границы определения лежат в точках пересечения графика  $\tau_r = 1$  с каждым из трёх графиков зависимостей  $\tau_{2r} = F(k, m = \text{const})$ . Нижние части графиков  $\tau_{2r}$  ( $\tau_{2r} \leq \tau_r = 1$ ) соответствуют внутреннему касанию составной ГПВ<sup>2r</sup> к гармонической ГПВ<sup>r</sup>, поэтому (18) даёт  $\tau_{2r} \leq \tau_r$  или результат, имеющий только математический, но не физический смысл. Энтропия во всех примерах  $H \geq 0$ , что соответствует двум этим смыслам и позволяет использовать энтропию для индикации рассогласования параметров сигналов.

**6. Общее свойство полученных моделей дискретизации времени.** Свойство заключается в получении одних моделей ( $\tau_{rл}$  из  $\tau_{2r}$ ;  $\tau_r$  из  $\tau_{rл}$ ) путем задания граничных значений коэффициентов согласования параметров  $k, m$  сигнала.

Для проверки общности моделей (11) и (5) зададим  $k = 1$  в (11) или выполним условие согласования. Получим  $T_{rл}(k = 1) = T_r$ , что позволяет сформулировать свойство:

Свойство 2: *Математическая модель гармонической полуволны с согласованными временными параметрами сигналов (процессов) является частным случаем модели составной полуволны, согласуемой двумя симметричными отрезками касательных прямых [9].*

Найдём предел выражения (18)  $\lim_{m \rightarrow 1} \tau_{2r}$  при  $(m \rightarrow 1)$ :

$$\lim_{m \rightarrow 1} \tau_{2r} = 1 + \frac{2}{\pi} \left[ \frac{1}{k} \left( \frac{1-k^2}{0} \right)^{1/2} \cdot 0 - \arccos(k) \right], \quad (19)$$

имеем неопределённость 0/0, которую раскроем по правилу Лопиталя:

$$\lim_{m \rightarrow 1} \frac{\arccos(m)}{(1-m^2)^{1/2}} = \lim_{m \rightarrow 1} \frac{(\arccos(m))^{(1)}}{((1-m^2)^{1/2})^{(1)}} = \lim_{m \rightarrow 1} \frac{-1}{-m} = 1.$$

Подставим  $\lim[\arccos(m)/(1-m^2)^{1/2}] = 1$  в (19), получим выражение (11). Из этого следует свойство:

Свойство 3: *Математическая модель составной полуволны с согласованными временными параметрами сигналов (процессов) при помощи двух отрезков прямых линий является частным случаем модели составной полуволны, согласуемой двумя симметричными частями гармонической функции [9].*

На основе Свойств 2 и 3 можно считать общим доказательством для двух типов согласования информационных параметров сигналов процессов доказательство теоремы. Применим энтропию по Хартли [26], для математической индикации согласованности и рассогласованности конкретного конечного множества параметров сигнала:  $\log_2(N=1)=0$ , или  $\log_2(N>1)>0$ . Используем особенность этих формул реагировать на величину коэффициента рассогласования скорости сигнала  $k = 1$  или  $k \leq 1$ . При этом вероятность исходов для такой индикации не важна. Коэффициент  $k$  может принимать любое значение в диапазоне  $0 \leq k \leq 1$ , так как причины, влияющие на него, зависят от конструкции, технических характеристик, процессов функционирования, сигналов конкретного технического объекта управления. Характеристики объектов, от которых зависит их нормальное функционирование, подтверждаются испытаниями, являются стабильными в течение всего интервала времени пригодности к эксплуатации, поэтому для конкретной серии изделий желательна малая и равновероятная интенсивность отказов составных частей. Кроме этого энтропия по Хартли позволяет получить универсальный инструмент индикации рассогласования параметров сигналов для неограниченного класса объектов управления систем реального времени. Для определения связи энтропии и величины интервала дискретизации  $\tau_{2r}$  (18) докажем необходимую теорему. Примем  $N \sim \tau_{2r}$ , так как именно  $\tau_{2r} = F(k, m)$  в (18) определяет количество вариантов рассогласованных составных траекторий, каждой из которых соответствует множество величин коэффициентов согласования  $k, m$ , поэтому запишем:

*Теорема об энтропии параметров сигналов процессов управления:* «Величины энтропии  $H_{гн}\{\Delta, V_A, a\} = \log_2(\tau_{гн}) > 0$  и  $H_{2r}\{\Delta, V_A, a\} = \log_2(\tau_{2r}) > 0$  рассогласованных максимальных параметров  $V_A < (\Delta a)^{1/2}$  сигнала  $p(t)$  процесса больше энтропии  $H_r\{\Delta, V, a\} = \log_2(\tau_r) = 0$  изначально согласованных параметров  $V = (\Delta a)^{1/2}$ . Величины  $H_{гн} > 0$  и  $H_{2r} > 0$  является индикаторами увеличения интервалов дискретизации времени  $\tau_{гн} \geq \tau_r$  и  $\tau_{2r} \geq \tau_r$ ». Теорема поясняется рисунками 6 и 7.

Доказательство. По определению из (5)  $\tau_r = T_r/T_i = 1$ , соответственно:  $H_r\{\Delta, V, a\} = \log_2(\tau_r = 1) = 0$ . По определению из (11)  $\tau_{гн} = T_{гн}/T_r$ , где  $T_{гн} > T_r$ , соответственно:  $H_{гн}\{\Delta, V_A, a\} = \log_2(\tau_{гн} > 1) > 0$ . По определению из выражения (18)  $\tau_{2r} = T_{2r}/T_r$ , где  $T_{2r} > T_r$ , соответственно:  $H_{2r}\{\Delta, V_A, a\} = \log_2(\tau_{2r} > 1) > 0$ . Учтём общие Свойства 2 и 3, что позволяет считать доказательство общим для трёх типов согласования информационных параметров сигналов процессов.



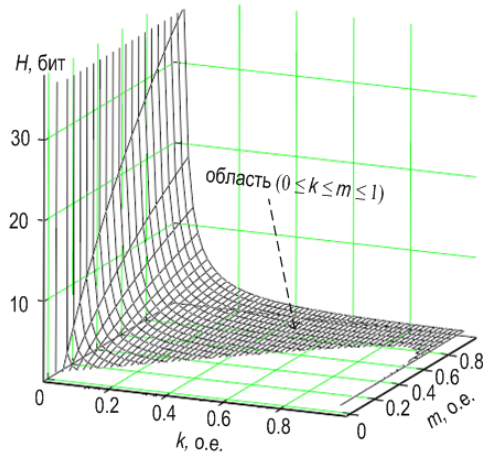


Рис. 7. Энтропия  $H_{2r} > 0$  параметров сигнала процесса в зависимости от величин коэффициентов согласования скоростей  $k, m$

Примечание. «На рисунках 6 и 7 отсутствуют отрицательные участки энтропии, так как им соответствует расположение графиков согласующих гармонических функций внутри графика согласуемой гармонической функции (полуволна с меньшим интервалом касается полуволны с большим интервалом внутри последней), что соответствует уменьшению интервала  $\tau_{2r} \leq \tau_r = 1$  и искажает физический смысл модели».

Преобразуем неравенство (17), учтём:  $m > 0, k > 0$ , получим  $m \geq k$ .

Свойство 4. Величина интервала равномерной дискретизации  $T_{2r}$  сигнала процесса с рассогласованными временными параметрами  $\{\Delta, V_A, a\}$  (при  $V_A < V = (\Delta a)^{0.5}$ ) всегда больше интервала дискретизации  $T_r$  сигнала процесса  $T_{2r} > T_r$  с согласованными временными параметрами  $\{\Delta, V, a\}$  (при  $V = (\Delta a)^{1/2}$ ). Индикатором рассогласования временных параметров сигнала (процесса)  $\{\Delta, V_A, a\}$  является выполнение неравенства  $H_{2r} > 0$  или  $m > k$ .

Графической иллюстрацией неравенства  $m \geq k$ , является энтропия коэффициентов  $k, m$  по выражению (18)  $H_{2r} = \log_2(\tau_{2r})$ , при этом в моделирующем пакете задано условие  $H_{2r} = \log_2(\tau_{2r}) \geq 0$  исходя из теоремы доказанной ранее. На рисунке 7 в системе координат  $k, m$ ,  $H_{2r}$  изображена поверхность, в которой находится область  $0 \leq k \leq m \leq 1$ . В этой области функция  $H_{2r} = \log_2(\tau_{2r})$  определена с доказанными ограничениями.

**7. Применение моделей дискретизации времени для описания скольжения (юза) поездов.** Явление юза или скольжения

поездов по рельсам при экстренном торможении особенно проявляется при дождливой погоде, заморозках и на спусках рельефа пути. Управление высокоскоростными поездами в этих условиях задаётся Европейской системой управления поездами (European Train Control System (ETCS)) [7, 8]. От начала торможения до останова поезда могут происходить четыре различных типа движения поезда, модели которых получим посредством преобразований выражений [25]:

- скольжение без изменения ускорения (замедления) или с постоянной скоростью, с последующим началом окончательного замедления и останова поезда и скольжение без начала замедления и останова поезда (авария);

- скольжение с изменением ускорения (замедления) или с переменной скоростью, с последующим началом окончательного замедления и останова поезда и скольжение с замедлением недостаточным для останова поезда (авария).

В статье [25] предложена новая концепция и методика построения математических моделей кривых экстренного торможения отличных от кривых ETCS и основанных на гармонических полуволнах. Показано, что кривые торможения ETCS описываются известными степенными полуволнами второго порядка. Гармонические полуволны в сравнении со степенными полуволнами позволяют объяснить особенности ETCS, получить предложения по их устранению. Эти полуволны применимы для разработки новых кривых экстренного торможения, которые позволяют плавно осуществлять экстренное торможение поездов. Составные гармонические полуволны используем для моделирования скольжения высокоскоростных поездов при экстренном торможении, как пример возможностей полученных математических моделей в практическом применении.

**7.1. Модель скольжения без изменения ускорения (замедления) или при постоянной скорости.** Для получения математического описания модели используем составную гармоническую полуволну согласуемую двумя отрезками прямых линий (рисунок 2). Скольжение поезда или перемещение с постоянной скоростью  $V_A = \text{const}$  происходит на интервале времени  $T_{\text{СКГЛ}} = t_A + |t_{k0}|$ , что следует из рисунка 2, для конкретизации которого используем (10),  $|t_{k0}|$ , применим  $\omega = \pi/T_{\Gamma}$ , относительное время скольжения запишем в виде выражения, которому соответствует рисунок 8:

$$\tau_{\text{СКГЛ}} = \frac{T_{\text{СКГЛ}}}{0.5T_{\Gamma}} = \frac{2}{\pi} \left( \frac{1}{k^2} - 1 \right)^{1/2}. \quad (20)$$

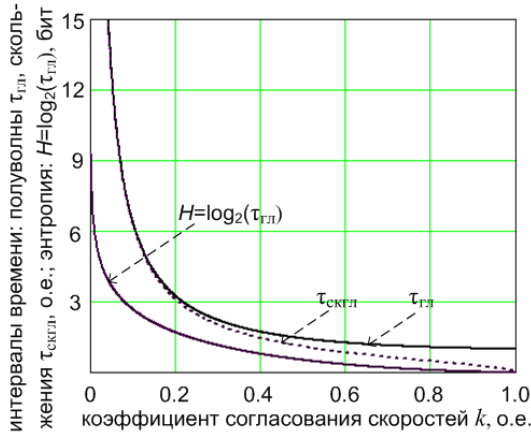


Рис. 8. Три характеристики скольжения по составной гармонической полуволне с постоянной скоростью

На рисунке 8, кроме относительного интервала времени скольжения поезда  $\tau_{скгл}$ , (20) даны: траектория торможения с постоянной скоростью скольжения на начальном участке  $\tau_{сл}$  (11), признак скольжения в виде «положительной энтропии параметров  $H_{сл} = \log_2(\tau_{сл}) \geq 0$ » для пояснения использования этого признака.

**7.2. Модель скольжения с увеличением замедления или при переменной скорости.** Для получения математического описания модели используем составную гармоническую полуволну согласуемую двумя отрезками гармоник (рисунки 4 и 9).

Скольжение поезда или перемещение с переменной скоростью  $V_A = var$  происходит на интервале времени  $T_{ск2г} = t_A + |t_{к0}|$ , что следует из рисунка 4, для конкретизации которого из (12) запишем:  $t_A = (1/\omega_k)(\arccos(m) - \varphi_0)$ ,  $|t_{к0}| = |\varphi_0/\omega_k|$ ,  $\omega_k = 2\pi/T_k$ , из (13) при  $r = T_k/2T_r$  применим (17). Относительное время скольжения запишем в виде выражения, которому соответствует рисунок 9.

$$\tau_{ск2г} = \frac{T_{ск2г}}{0.5T_r} = \frac{2}{\pi} \frac{m}{k} \left( \frac{1-k^2}{1-m^2} \right)^{1/2} \cdot \arccos(m), \quad (21)$$

где выполняются условия:  $k \leq m$  из (18) и  $\tau_{ск2г} \geq 1$ .

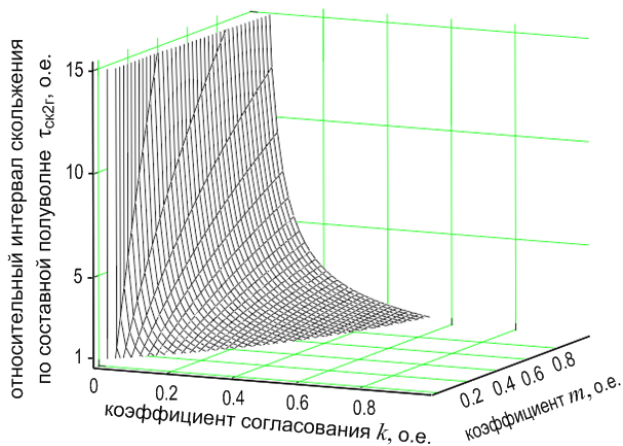


Рис. 9. Относительный интервал скольжения по составной полуолне с переменной скоростью

При этом индикатором режима скольжения (юза) является теорема об энтропии параметров сигналов процессов управления, величина энтропии  $H_{2г}\{\Delta, V_A, a\} = \log_2(\tau_{2г}) > 0$ , поясняется рисунками 7 и 9.

**7.3. Модели скольжения без достаточного замедления для своевременного останова.** Аварийное скольжение поезда (когда авария возможна) может начинаться при постоянной и переменной скорости, когда поезд переходит точку  $A$  (20) с постоянной скоростью скольжения  $V_A = \text{const}$  (рисунок 2) или перемещается с переменной скоростью  $V_A = \text{var}$  (21) через точку  $A$  (рисунок 4). При этом замедление поезда не достигает необходимой величины для останова в точке с координатой пути ( $\Delta$ ) и движение продолжается до точки с координатой  $p(t_C)$  (рисунки 2, 4).

**Модель аварийного скольжения на постоянной скорости.** Из прямоугольного треугольника на рисунке 2 с тремя координатами вершин  $(-t_{к0}, 0)$ ,  $(0.5T_{г}, p(t_C))$ ,  $(0.5T_{г}, 0)$  запишем координату точки возможной аварии:  $p(t_C) = V_A(t_{к0} + 0.5T_{г})$ . Воспользуемся из (6) выражением  $0.5T_{г} = 0.5\pi\Delta/V$ ,  $t_{к0}$  из (10),  $V_A = kV$ , получим:

$$\frac{p(t_C)}{\Delta} = (1 - k^2)^{1/2} + k \arcsin(k), \quad (22)$$

где при  $k = 1$ ,  $p(t_C) = 0.5\Delta\pi$ . С учётом Свойства 3 график по выражению (22), как частный случай общей математической модели,

совместим с графиком модели аварийного скольжения поезда на переменной скорости далее по тексту.

**Модель аварийного скольжения на переменной скорости.**

В выражение (12) подставим  $t = 0.5T_{\Gamma}$ , то есть найдём точку пересечения оси симметрии гармонической полуволны ГПВ<sup>r</sup>  $t_C = 0.5T_{\Gamma}$  с ГПВ<sup>2r</sup> (рисунок 4). При этом используем  $\omega_k = 2\pi/T_k$  (12),  $T_k/2T_{\Gamma} = r$ , коэффициент  $r$  запишем из (17), фазу  $\phi_0$  из (14), применим известную формулу  $[0.5\pi - \arccos(x) = \arcsin(x)]$ , выражение  $\Delta_k/\Delta$  (16), получим:

$$\frac{p(t_C)}{\Delta} = \left( \frac{1-k^2}{1-m^2} \right)^{1/2} \cdot \sin \left[ \frac{k}{m} \left( \frac{1-m^2}{1-k^2} \right)^{1/2} \cdot \arcsin(k) + \arccos(m) \right], \quad (23)$$

где  $p(t_C)/\Delta \geq 1$ , что характерно для аварийной ситуации.

Выражения (22) и (23) иллюстрируются общим рисунком 10. На рисунке 10 изображена поверхность в относительных единицах измерения. В системе координат  $\{k, m, p(t_C)/\Delta\}$  находится поверхность в виде совокупности точек с координатами, в которых происходят аварийные ситуации из-за эффекта скольжения.

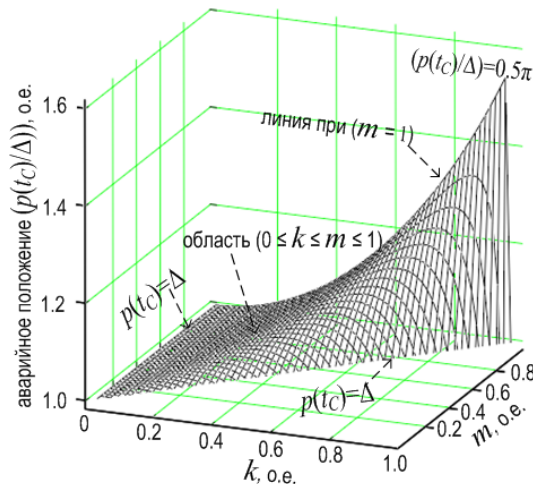


Рис. 10. Относительное аварийное положение поезда при скольжении через точки с координатами  $(t_A, p(t_A))$  и  $(0.5T_{\Gamma}, \Delta)$  в точку  $(0.5T_{\Gamma}, p(t_C))$

Границы поверхности отражают конкретные ситуации и свойства скольжения состава по составной гармонической полуволне:

1) При  $k=m \neq 1$  из (23) имеем  $p(t_C)/\Delta = 1$ , что на рисунке 10 соответствует границе, проходящей параллельно диагонали плоскости  $k, m, p(t_C)/\Delta$ .

2) При  $k=m=1$  из (23) и (22) имеем  $\max p(t_C) = 0.5\Delta\pi$ . Математические модели составных гармонических полуволн с согласованными временными параметрами сигналов процессов, применяемые для моделирования аварийной ситуации от скольжения поезда дают ограничение расстояния скольжения.

3) При  $k=0, m \rightarrow 1$  из (23), по правилу Лопитала:  $p(t_C)/\Delta = 1$ , что на рисунке 10 соответствует границе, проходящей параллельно оси координат  $0m$  в системе координат  $k, m, p(t_C)/\Delta$ .

4) Выражение (23) перевести в выражение (22) при  $m=1$  прямыми преобразованиями не удаётся, поэтому применим преобразование бесконечно малой высшего порядка  $0(m)$ , чем величина  $m$ . Зададимся  $(1-m^2)=0(m)=0.00010$ , откуда  $m=(1-0.00010)^{1/2}=0.99995$ . Подставим значения величин  $0(m)$  и  $m$  в (23), наложим графики (23) и (22) и убедимся в их полном совпадении. Иллюстрацией этого совпадения графиков является рисунок 11.

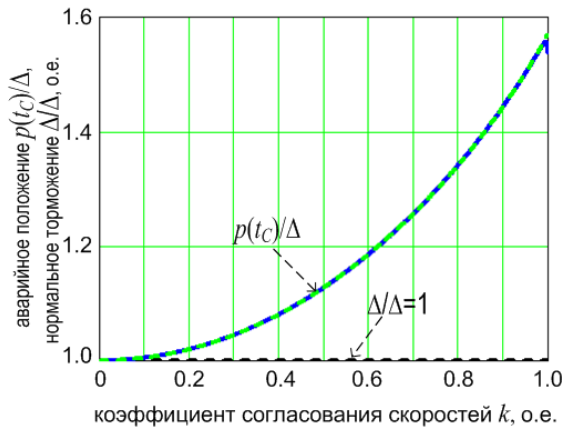


Рис. 11. Совпадение относительных координат аварийного положения поезда при скольжении с постоянной и переменной скоростью при  $m=1$

**8. Пример применения моделей дискретизации времени для расчёта скольжения (юза) высокоскоростных поездов в режиме экстренного торможения.** Применим три рассмотренных и описанных модели для моделирования скольжения при одинаковой величине пути торможения  $\Delta$ , но для различных погодных

и рельефных условий на участке торможения, которые задаются коэффициентами  $k \leq m < 1$ . Используем математический пакет MathCAD. Для пояснения рассмотрим таблицу 1 в основных ситуациях.

**Строка 3.** Предположим, что по сухим рельсам, на равнине высокоскоростной поезд осуществляет экстренное торможение по половине гармонической полуволны (1), [25], где  $0 \leq t \leq 0.5T_r$ .

Таблица 1. Примеры расчётов параметров скольжения (юза) поезда

Тип гармонической полуволны, индекс «...»	$k$ , о.е.	$m$ , о.е.	$\Delta$ , м	$V_{г,гл,2г}$ , м/с	(№), $H = \log_2(\tau)$ , бит	(№), $0.5T_{г,гл,2г}$ , с	(№), $T_{ск}$ , с	(№), $p(t)$ , М
Модели	Параметры движения поезда				Модели и результат			
Согласованная, индекс «г»	1	1	1500	138.89	(5), (11), $\log_2(\tau_r)=0$	(5), 16.96	нет юза	нет аварии
Согласуемая прямой линией, индекс «гл»	0.5	1	1500	69.45	(11), $\log_2(\tau_{гг})=0.522$	(11), 24.36	(20), 18.71	(22), 1692.0 авария
Согласуемая гармонической кривой, инд. «2г»	0.5	0.8	1500	86.81	(18), $\log_2(\tau_{2г})=0.355$	(18), 18.34	(21), 16.05	(23), 1655.0 авария

При начальной скорости движения высокоскоростного поезда в режиме торможения  $V=500$  км/ч=138.89 м/с, необходимо обеспечить тормозной путь  $\Delta=1500$  м с замедлением  $a=V^2/\Delta=12.86$  м/с<sup>2</sup>. Время торможения (5) при заданных условиях  $0.5T_r=16.96$  с, скольжение отсутствует. Авария не прогнозируется, так как скорость  $V_A$ , при которой должно начаться торможение  $V_A=V$ , то  $k=V_A/V=1$  при  $t=0$  (рисунок 1). Проверим энтропию параметров движения на их согласованность:  $H_r = \log_2(\tau_r=1)=0$  (строку 3 таблицы 1). Условие согласования параметров  $\{\Delta, V, a\}$  модели:  $V^2=\Delta a$ .

**Строка 4.** Предположим, что по обледеневшим рельсам, на уклоне высокоскоростной поезд осуществляет экстренное торможение по половине гармонической полуволны, согласуемой отрезком прямой линии (11), [25], где  $\tau_{гг} = T_{гг}/T_r$ ,  $0 \leq t \leq 0.5T_{гг}$ . При этом только в момент времени  $t_A$  тормоза поезда создают необходимую величину замедления, максимум которого наступает в конце тормозного пути  $a=V^2/\Delta=12.86$  м/с<sup>2</sup> в точке с координатами  $(0.5T_{гг}, \Delta)$ , рисунок 2. Время торможения (11) при заданных условиях скольжения  $0.5T_{гг}=24.36$  с. Чтобы обеспечить заданный путь торможения в  $\Delta=1500$  м необходимо

уменьшить начальную скорость торможения  $V=500$  км/ч в два раза  $V_A=kV=69.45$  м/с  $=250$  км/ч (рисунок 2 в точке с координатами:  $-t_{к0}$ ,  $p(-t_{к0})=0$ ). Проверим энтропию параметров движения на их согласованность:  $H_{гп}=\log_2(\tau_{гп})=0.522$  бит  $>0$  (строку 3 таблицы 1). Условие согласования параметров  $\{\Delta, V, a\}$  модели не выполняются:  $V_A^2 \neq \Delta a = V^2$ . Время скольжения найдём из (20) при  $k=0.5$   $T_{скгл}=18.71$  с. Авария в виде столкновения с предыдущим поездом возможна при превышении расстояния  $p(t_C) \geq 1692.0$  м, которое больше чем тормозной путь  $\Delta=1500$  м. Это произойдёт, если в момент времени  $t_A$  тормоза не начнут эффективно работать (рисунок 2).

**Строка 5.** Торможение, совмещенное со скольжением, приводит к уменьшению замедления на интервале времени  $-t_{к0} \leq t \leq t_A$ . В момент времени  $t_A$  тормоза начинают эффективно действовать. Энтропия параметров  $H_{гп}=\log_2(\tau_{гп})=0.355$  бит  $>0$ . Время торможения (18) при заданных условиях скольжения  $0.5T_{2г}=21.70$  с. Начальную скорость торможения  $V=500$  км/ч, при идеальных условиях, необходимо в данном случае уменьшить в 1.6 раза  $V_{к0}=(k/m)V=86.81$  м/с  $=312.52$  км/ч. Это следует из двух равенств  $V_A=kV$  и  $V_A=mV_{к0}$  (16). Учёт условий погоды и рельефа пути должна осуществлять робастная система управления скоростным движением, с применением теоремы об энтропии параметров сигналов процессов управления. При этом только предварительное снижение скорости  $V_{к0}$  высокоскоростного поезда на скользких рельсах и при уклонах рельефа позволяют избежать аварийной ситуации или сохранить постоянной величину тормозного пути  $\Delta$ .

**9. Заключение.** Сформулирована и исследована гипотеза подобия, в соответствии с которой исследование и применение сигналов управляемых процессов в робастных системах управления заменено исследованием и применением гармонических полуволн. Предварительно, при обследовании объекта управления, замерены максимальные величины временных параметров сигнала и полуволн: скорость, ускорение и резкость, которые используют для описания гармонических полуволн, так как параметры спектра отсутствуют. Гармонические полуволны разделены на две группы: согласованные и согласуемые полуволны. Для определения степени согласования применена энтропия временных параметров гармонических полуволн. Обнаружено свойство увеличения интервалов дискретизации времени сигнала совместно с ростом энтропии его параметров, доказана теорема об энтропии параметров сигналов процессов управления. В качестве примера теорема применена для построения математической модели торможения высокоскоростного состава



со скольжением (юзом). Модель работает в двух вариантах скольжения: с постоянной максимальной начальной скоростью и переменной скоростью, что необходимо для управления составом в реальных климатических условиях и с учетом рельефа пути. В дальнейшем возможно применение теоремы для обнаружения информационных атак на системы ГЛОНАСС и GPS спутниковой навигации беспилотных высокоскоростных поездов и беспилотных летательных аппаратов.

### Литература

1. Коберниченко В.Г. Основы цифровой обработки сигналов // Учеб. Пособие. Екатеринбург: Изд-во Урал. ун-та, 2018. 150 с.
2. Курячий М.И., Гельчер А.А. Цифровая обработка сигналов // Томск: Изд-во Томск. Гос. ун-т, 2018. 234 с.
3. Контрольно-измерительное оборудование R&S 2019. Каталоги Rohde&Schwarz. 2019. 256 с. URL: [https://cdn.rohde-schwarz.com/ru/downloads\\_45/common\\_library\\_45/interactive\\_catalog\\_2018\\_19\\_ru.pdf](https://cdn.rohde-schwarz.com/ru/downloads_45/common_library_45/interactive_catalog_2018_19_ru.pdf) (дата обращения: 12.03.2020).
4. Infrastructure Status u-entwicklungsbericht 2019 // Deutsche Bahn. 2020. 124 p.
5. ERTMS Level 3: the Game- Chander. 2022. Programme technical paper. London.
6. Thales consortium to install ECTS Level 1 ATR system in Thailand // Railway Technology. 2019. 20 p. URL: <https://www.railway-technology.com/news/thales-consortium-to-install-etsc-level-1-atp-system-in-thailand/> (дата обращения: 12.03.2020).
7. Ehret Marc. Virtual Train Brakes. Eisenbahnwesen-Seminar // TU-Berlin. 2020. 49 p. URL: <https://www.static.tu.berlin/fileadmin/www/10002264/ews/2020-sosc/2020-05-11-folien.pdf> (дата обращения: 12.03.2020).
8. Introduction to ETCS Braking Curves. ERA ERTMS Unit. Version 1.5. 2020. 28 p.
9. Майоров Б.Г. Свойства гармонических и составных полувольт, определение интервала равномерной дискретизации времени цифровых сигнальных процессоров // Информатика и автоматизация. 2022. Т. 21. № 1. С. 95–125. DOI: 10.15622/ia.2022.21.4.
10. Лазарев В.Л. Информационное шкалирование вариаций законов распределения параметров в приложениях к задачам мониторинга и управления // Научно-технические ведомости СПб ГПУ. Информатика. Телекоммуникации. Управление. 2019. Т. 12. № 2. С. 7–15.
11. Лазарев В.Л., Спасивцев А.В., Захаров В.В. Оценка видов законов распределений на основе величины энтропийного коэффициента // Изв. Вузов. Приборостроение. 2020. Т. 63. № 2. С. 113–119.
12. Шамолин М.В. Алгоритмы диагностирования в некоторых системах прямого и непрямого управления // Итоги науки и техники. Серия «Современная математика и ее приложения. Тематические обзоры. 2022. Т. 205. С. 107–118. DOI: 10.36535/0233-6723-2022-205-107-118.
13. Даринцев О.В., Мигранов А.Б. Аналитический обзор подходов к распределению задач в группах мобильных роботов на основе технологий мягких вычислений // Информатика и автоматизация. 2022. Т. 21. № 4. С. 729–757. DOI: 10.15622/ia.21.4.4.

14. Валиков П.И., Мусатов Р.Л., Степанов Б.В. Электрогидравлическая система управления движением робототехнического комплекса // Электронные информационные системы. 2020. № 2(25). С. 62–68.
15. Петренко В.И., Тебуева Ф.Б., Гурчинский М.М., Антонов В.О., Павлов А.С. Прогнозная оценка траектории руки оператора для решения обратной задачи динамики при копирующем управлении // Труды СПИИРАН. 2019. Т. 18. № 1. С. 123–147. DOI: 10.15622/sp.18.1.123-147.
16. Григорьев И.С., Данилина И.А. Оптимизация траектории перелётов космических аппаратов с дополнительным топливным баком. II. Автоматика и телемеханика. 2018. № 2. С. 135–153.
17. Manyam S.G. Optimal dubins paths to intercept a moving target on a circle // Proc. of the American Control Conference. 2019. pp. 828–834.
18. Майоров Б.Г. Способ непрерывной оптической связи с низколетающей целью // Патент RU №2 715 499 С1. 2020. МПК G 01 S 13/66.
19. Баунин В.Г., Землеханов А.Р., Швецов Н.В., Аверин Д.А. Система управления двухосным поворотным устройством оптико-электронной системы сопровождения целей // Электронные информационные системы. 2020. № 2(25). С. 43–49.
20. Сенкевич Ю.И., Марапулец Ю.В., Луковенкова О.О., Солодчук А.А. Методика выделения информативных признаков в сигналах геоакустической эмиссии // Труды СПИИРАН. 2019. Т. 18. № 5. С. 1066–1092.
21. Хлистунов В.Н. О погрешности интерполяции дискретных методов измерения // Приборостроение. 1960. № 5. С. 3–5.
22. Майоров Б.Г. Обобщенный критерий наибольшего отклонения входных сигналов систем управления // Автоматика и телемеханика. 2005. № 10. С. 148–155.
23. Майоров Б.Г. Восстановление сигнала в системах реального времени по равномерным выборкам с уменьшением интервала Найквиста // Системы и средства информатики. 2019. Т. 29. № 2. С. 95–112.
24. Nyquist H. Certain topics in telegraph transmission theory // Transactions of the American Institute of Electrical Engineers. 1928. vol. 47. no. 2. pp. 617–644.
25. Майоров Б.Г. Применение гармонических полуволн для автоматизации управления высокоскоростными поездами // Информатика и автоматизация. 2023. Т. 22. № 6. С. 1387–1414. DOI: 10.15622/ia.22.6.5.
26. Hartley R.V.L. Transmission of Information // Bell System Technical Journal. 1928. vol. 7. no. 3. pp. 535–563.

**Майоров Борис Геннадьевич** — канд. техн. наук, ведущий научный сотрудник, центр специальных систем, Акционерное общество «Научно-производственное предприятие «Рубин»; доцент, Пензенский государственный технологический университет. Область научных интересов: цифровая обработка сигналов, системы сбора и отображения информации в реальном времени. Число научных публикаций — 88. [bgmaurov@yandex.ru](mailto:bgmaurov@yandex.ru); улица Байдукова, 2, 440015, Пенза, Россия; р.т.: +7(8412)204-777.

**Поддержка исследований.** Работа выполнена по плану научно-технического развития предприятия на 2024 год.

B. MAYOROV

**MODELS OF COMPOSITE HARMONIC HALF-WAVES AND THE RELATIONSHIP OF TIME SAMPLING WITH THE ENTROPY OF TIME PARAMETERS OF SIGNALS**

*Mayorov B. Models of Composite Harmonic Half-Waves and the Relationship of Time Sampling with the Entropy of Time Parameters of Signals.*

**Abstract.** The problem of finding the absolute error of stepwise and linear interpolation of the control signal from uniform samples from it using models of composite harmonic half-waves is solved. Previously, during the inspection of the control object, the maximum values of the signal parameters and half-waves are determined: speed, acceleration and sharpness, there are no spectrum parameters. To determine the values of the intervals of uniform sampling of time, two groups of models of "harmonic half-waves" are considered. The first group of models is described by harmonic time functions whose parameters are consistent. The second group of models is described by composite harmonic functions of time, thereby the time parameters of the signals are consistent. It is proved that with an increase in the entropy of the maximum values of the signal parameters, the value of the time sampling interval increases without increasing the interpolation error. Thus, the entropy value of the signal parameters serves as an indicator of their inconsistency. The results of modeling and graphs obtained in the environment of the mathematical package are presented. The results are intended to optimize the loading of input tasks and primary information processing of processors in robust real-time automation systems, for example, used to control high-speed trains when braking in sliding or skidding mode.

**Keywords:** robust systems, real time, time sampling, zero- and first-order interpolation, composite, harmonic functions, entropy of signal parameters, braking model with high-speed trains.

**References**

1. Kobernichenko V.G.. Osnovy tsyvrovoy obrabotki signalov. Uchebnoe posobie. [Fundamentals of digital signal processing. Textbook]. Ekaterinburg: Ural University Press, 2018. 150 p. (In Russ.).
2. Kuryachiy M.I., Gelcher A.A. Tsyvrovaya obrabotka signalov [Digital Signal Processing] Tomsk: Tomsk State University Publishing House, 2018. 234 p. (In Russ.).
3. Kontrol'no-izmeritel'noe oborudovanie R&S 2019. Katalogi Rohde&Schwarz [Test equipment R&S 2019. Catalogs Rohde&Schwarz] 2019. 256 p. Available at: [https://www.rohdeschwarz.com/ru/top\\_topics\\_russia/catalogs/\\_231625.html](https://www.rohdeschwarz.com/ru/top_topics_russia/catalogs/_231625.html) (accessed: 12.03.2020).
4. Infrastructure Status u-entwicklungsbericht 2019. Deutsche Bahn. 2020. 124 p.
5. ERTMS Level 3: the Game-Changer. 2022. Programme technical paper. London.
6. Thales consortium to install ECTS Level 1 ATR system in Thailand. Railway Technology. 2019. 20 p. Available at: <https://www.railway-technology.com/news/thales-consortium-to-install-etc-level-1-atp-system-in-thailand/> (accessed: 12.03.2020).
7. Ehret Marc. Virtual Train Brakes. Eisenbahnwesen-Seminar. TU-Berlin. 2020. 49 p. Available at: <https://www.static.tu.berlin/fileadmin/www/10002264/ews/2020-sose/2020-05-11-folien.pdf> (accessed: 12.03.2020).
8. Introduction to ETCS Braking Curves. ERA ERTMS Unit. Version 1.5. 2020. 28 p.

9. Mayorov B.G. [Properties of harmonic and composite half-waves, determination of the interval of uniform time sampling of digital signal processors]. *Informatika i avtomatizatsiya – Informatics and Automation*. 2022. vol. 21. no. 1. pp. 95–125. DOI: 10.15622/ia.2022.21.4. (In Russ.).
10. Lazarev V.L. [Information scaling of variations in the laws of distribution of parameters in applications to monitoring and control problems]. *Nauchno-tekhnicheskie vedomosti St. Petersburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie - Scientific and technical statements of the St. Petersburg State Polytechnic University. Computer science. Telecommunications. Control*. 2019. vol. 12. no. 2. pp. 7–15. (In Russ.).
11. Lazarev V.L., Spesivtsev A.V., Zakharov V.V. [Evaluation of the types of distribution laws based on the value of the entropy coefficient]. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie – Izvestiya of higher educational institutions. Instrumentation*. 2020. vol. 63. no. 2. pp. 113–119. (In Russ.).
12. Shamolin M.V. [Diagnostic algorithms in some systems of direct and indirect control. Results of science and technology]. *Itogi nauki i tekhniki. Seriya Sovremennaya matematika i ee prilozheniya. Tematicheskii obzor – Results of science and technology. Series Modern Mathematics and Its Applications. Subject Review*. 2022. vol. 205. pp. 107–118. DOI: 10.36535/0233-6723-2022-205-107-118. (In Russ.).
13. Darintsev O.V., Migranov A.B. [Analytical review of approaches to the distribution of tasks in groups of mobile robots based on soft computing technologies]. *Informatika i avtomatizatsiya – Informatics and Automation*. 2022. vol. 21. no. 4. pp. 729–757. DOI: 10.15622/ia.21.4.4. (In Russ.).
14. Valikov P.I., Musatov R.L., Stepanov B.V. [Electrohydraulic motion control system for a robotic complex]. *Elektronnye informatsionnye systemy – Electronic information systems*. 2020. no. 2(25). pp. 62–68. (In Russ.).
15. Petrenko V.I., Tebueva F.B., Gurchinsky M.M., Antonov V.O., Pavlov A.S. [Predictive estimation of the trajectory of the operator's hand for solving the inverse problem of dynamics under copy control]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2019. vol. 18. no. 1. pp. 123–147. DOI: 10.15622/sp.18.1.123-147. (In Russ.).
16. Grigoriev I.S., Danilina I.A. Optimising flight for space vehicles with an additional fuel tank. II. *Autom. Remote Control*. 2018. vol. 79. no. 2. pp. 311–326.
17. Manyam S.G. Optimal dubins paths to intercept a moving target on a circle. *Proc. of the American Control Conference*. 2019. pp. 828–834.
18. Mayorov B.G. [The method of continuous optical communication with a low-flying target. Patent RU No. 2715499 C1. 2020. IPC G01S13/66]. *Bulletin of Inventions No. 7*. 2020. 2020. (In Russ.).
19. Baunin V.G., Zemlekanov A.R., Shvetsov N.V., Averin D.A. [Control system for a two-axis rotary device of an optical-electronic target tracking system]. *Elektronnye informatsionnye systemy - Electronic information systems*. 2020. no. 2(25). pp. 43–49. (In Russ.).
20. Senkevich Yu.I., Marapulets Yu.V., Lukovenkova O.O., Solodchuk A.A. [Technique for extracting informative features in geoacoustic emission signals]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2019. vol. 18. no. 5. pp. 1066–1092. (In Russ.).
21. Khlisunov V.N. [On the error of interpolation of discrete measurement methods]. *Priborostroenie – Instrumentation*. 1960. no. 5. pp. 3–5. (In Russ.).
22. Maiorov B.G. The generalized maximum deviation criterion for input signals of control systems // *Autom. Remote Control*. 2005. vol. 66. no. 10. pp. 1666–1672.
23. Mayorov B.G. [Signal recovery in real-time systems from uniform samples with decreasing Nyquist interval]. *Sistemy i sredstva informatiki – Systems and means of informatics*. 2019. vol. 29. no. 2. pp. 95–112. (In Russ.).

24. Nyquist H. Certain topics in telegraph transmission theory. Transactions of the American Institute of Electrical Engineers. 1928. vol. 47. № 2. pp. 617–644.
25. Mayorov B.G. [Application of Harmonic Half-waves for Automation of High-speed Trains Control]. Informatika i avtomatizatsiya – Informatics and Automation. 2023. vol. 22. no. 6. pp. 1387–1414. DOI 10.15622/ia.22.6.5. (In Russ.).
26. Hartley R.V.L. Transmission of Information. Bell System Technical Journal. 1928. vol. 7. no. 3. pp. 535–563.

**Mayorov Boris** — Ph.D., Leading researcher, Center for special systems, Joint Stock Company Scientific and Production Enterprise Rubin; Associate professor, Penza State Technological University. Research interests: digital signal processing, systems for collecting and displaying information in real time. The number of publications — 88. bgmayorov@yandex.ru; 2, Baidukova St., 440015, Penza, Russia; office phone: +7(8412)204-777.

**Acknowledgements.** The work was carried out according to the plan of scientific and technical development of the enterprise in 2024.

А.В. СУРЧЕНКО, Ю.А. НЕДБАЙЛО  
**МЕТОДИКА КОМПРЕССИИ ДАННЫХ В НАКРИСТАЛЬНЫХ  
И МЕЖПРОЦЕССОРНЫХ СЕТЯХ С ШИРОКИМИ КАНАЛАМИ  
И ПОЛИТИКОЙ УПРАВЛЕНИЯ ПОТОКОМ WORMHOLE**

*Сурченко А.В., Недбайло Ю.А. Методика компрессии данных в накристалльных и межпроцессорных сетях с широкими каналами и политикой управления потоком wormhole.*

**Аннотация.** Увеличение количества вычислительных ядер является одним из основных современных способов повышения производительности процессоров. При этом увеличивается и нагрузка на подсистему памяти процессора в связи с растущим числом инцидентов обращений в память. Одним из нестандартных подходов к повышению производительности подсистемы памяти является аппаратная компрессия данных, позволяющая, во-первых, повысить эффективный объем кэш-памяти, снижая частоту запросов в оперативную память, а во-вторых, снизить интенсивность трафика в подсистеме памяти за счет более плотной упаковки данных. В работе рассматривается применение аппаратной компрессии данных в сети-на-кристалле и межпроцессорных каналах связи в конфигурации с широкими каналами передачи данных и политикой управления потоком wormhole. Существующие решения для такой конфигурации нельзя считать применимыми, т.к. они принципиально основаны на использовании узких каналов передачи данных и политиках управления потоком, предполагающих передачу пакета в неразрывном виде, что может не соблюдаться при применении политики wormhole. Предлагаемая в работе методика позволяет использовать аппаратную компрессию для рассматриваемой конфигурации за счет переноса процесса компрессии и декомпрессии из самой сети в соединяемые устройства, а также ряда оптимизаций по сокращению задержек на преобразование данных. Рассматриваются оптимизации некоторых частных случаев передачи данных – передачи больших пакетов данных, состоящих из нескольких кэш-строк, а также нулевых данных. Особое внимание в работе уделено передаче данных по межпроцессорным каналам связи, в которых, в связи с их меньшей пропускной способностью по сравнению с сетью-на-кристалле, применение компрессии способно оказать наибольший эффект. Повышение пропускной способности подсистемы памяти при использовании в ней аппаратной компрессии данных подтверждается экспериментальными результатами, показывающими относительное увеличение IPC в задачах пакета SPEC CPU2017 до 14 процентов.

**Ключевые слова:** архитектура процессора, подсистема памяти, аппаратная компрессия данных, сеть-на-кристалле, межпроцессорные каналы связи, модель процессора.

**1. Введение.** В настоящее время одним из основных способов увеличения производительности процессоров общего назначения является увеличение числа входящих в них ядер. Этот подход позволяет увеличить количество задач или потоков, которые могут быть параллельно запущены на машине, как следствие, возрастает объем вычислений, который машина способна произвести за определенное время. Однако при этом возрастает нагрузка на подсистему памяти, поток запросов в которую увеличивается, а при

достаточно плотном потоке обращений ее пропускной способности может оказаться недостаточно. В связи с этим повышение пропускной способности подсистемы памяти становится одной из ключевых задач разработчиков современных процессорных систем [1 – 4].

Основные подходы к решению этой задачи включают в себя оптимизацию работы общей кэш-памяти для снижения частоты промахов в кэш и, как следствие, уменьшения частоты обращения в оперативную память, а также совершенствование топологии и архитектуры маршрутизаторов сети-на-кристалле процессора для повышения ее пропускной способности и снижения задержек на передачу.

Альтернативным подходом, уже достаточно длительное время обсуждаемым в научном сообществе, можно считать аппаратную компрессию данных. Ее применение позволяет преобразовать данные, как правило, на уровне кэш-строк, к меньшему размеру. За счет этого удается, с одной стороны, снизить трафик в подсистеме памяти [5 – 10], а с другой – увеличить эффективный объем кэш-памяти [11 – 13] или даже оперативной памяти [14 – 16] за счет того, что в памяти того же объема потенциально можно будет разместить большее количество кэш-строк.

Традиционные алгоритмы компрессии не являются подходящими для применения в аппаратуре микропроцессора, поскольку их использование может привести к появлению значительных дополнительных задержек по чтению данных, росту энергопотребления и нарушению структурной организации устройств. В связи с этим при создании алгоритмов, предназначенных для аппаратной компрессии данных, разработчики стараются учитывать эти факторы и минимизировать воздействие компрессии на параметры всей системы.

В данной работе рассматривается применение аппаратной компрессии при передаче данных, а именно в каналах сети-на-кристалле и в межпроцессорных каналах связи. Рассмотрим, как аппаратная компрессия данных может повлиять на факторы, которые ограничивают производительность этих каналов.

В отношении сети-на-кристалле необходимо рассмотреть ситуацию, когда в нее одновременно поступает большое число запросов от всех ядер системы. В таком случае в сети наблюдается повышенный трафик, и в ее узлах возрастает число конфликтных ситуаций, вызванных тем, что пакеты с нескольких входных направлений требуется передать в одном и том же выходном направлении. В ходе арбитража ресурс предоставляется только одному

из направлений, посылки с других направлений находятся в буфере вплоть до освобождения ресурса и их выбора арбитром. Именно это ожидание освобождения ресурса и является основной причиной ограничения производительности в сети-на-кристалле.

Использование аппаратной компрессии уменьшает размер пакета, в связи с чем уменьшается количество флитов – частей пакета, на которые он разбивается – а значит, передача одного пакета через узел занимает меньшее время. Как следствие, конфликтных ситуаций в сети становится меньше, а в тех случаях, когда они все же возникают, пакеты, проигравшие при арбитраже, будут проводить меньше времени ожидания в буфере. Таким образом, общая пропускная способность сети-на-кристалле возрастет.

Рассматривая межпроцессорные каналы связи, можно выделить несколько факторов, ограничивающих их пропускную способность. Во-первых, несмотря на порой более высокую частоту работы этих каналов, уже не привязанную к частоте работы процессора, их ширина ограничена количеством выводов процессора, выделенных под такую передачу. В связи с этим пропускная способность этих каналов в целом всегда ниже, чем сети-на-кристалле. Во-вторых, помимо пакетов с данными, по этим каналам передаются пакеты и других типов, соответствующих сообщениям системного протокола, связанным с поддержкой когерентности. Помимо пакетов системного протокола когерентности, по каналам могут также передаваться служебные сообщения, связанные с поддержкой помехоустойчивости, или необходимые для корректной обработки принимаемых данных. Однако за счет своих размеров пакеты с данными все еще будут составлять достаточно существенную долю сетевого трафика.

Из-за более низкого темпа передачи при высокой нагрузке на межпроцессорные каналы связи большая часть запросов будет помещена в буфер в ожидании передачи. Применение аппаратной компрессии позволит сократить объем передаваемых по каналам пакетов с данными, а значит, будет освобождено место для отправки дополнительных сообщений. За счет этого степень буферизации снизится, а пропускная способность межпроцессорных каналов связи возрастет [17].

## **2. Современное состояние. Актуальность исследования.**

Научные работы, связанные с аппаратной компрессией данных при их передаче, можно условно разделить на две группы – работы, посвященные используемым алгоритмам компрессии, и работы, связанные непосредственно с разработкой методик по осуществлению компрессии и взаимодействию со сжатыми данными.



В ходе рассмотрения алгоритмов компрессии в [18] они были разделены на несколько групп в зависимости от того, какой тип локальности данных используется алгоритмом для достижения компрессии. Авторы выделяют следующие типы локальности:

- локальность малых значений: отбрасываются нулевые или единичные разряды для малых положительных или отрицательных значений;
- локальность сгруппированных значений: используется слабое различие в значениях между находящимися по соседству данными;
- локальность изолированных значений: применяется кодирование часто встречающихся значений или последовательностей значений меньшим числом битов.

В одной из более поздних работ, посвященных алгоритмам компрессии, большое внимание уделено такому параметру алгоритма, как задержка декомпрессии [11]. Поскольку именно декомпрессия данных при работе алгоритма находится на критическом пути чтения данных, при разработке алгоритма необходимо добиться максимального сокращения затрачиваемого на нее времени.

Авторы статьи [11] предлагают свой алгоритм ВΔI (Base-Delta-Immediate), основанный на локальности сгруппированных значений и работающих на уровне кэш-строк. Исходная строка разбивается на сегменты равного размера, и затем значения этих сегментов сравниваются с некоторым базовым значением. В том случае, если разница в значениях для каждого из сегментов достаточно мала, в сжатой кэш-строке размещается базовое значение и набор разностей для восстановления исходных значений сегментов. За счет того, что исходные значения восстанавливаются параллельно для каждого из сегментов, скорость декомпрессии при использовании этого алгоритма очень высока. В соответствии с проведенными расчетами, при аппаратной реализации декомпрессия по алгоритму ВΔI укладывается в 1 такт при рабочей частоте 2 ГГц [18].

В данной работе используется модифицированная версия алгоритма, названная ВΔI\*-HL [19, 20]. Ее особенностями является модифицированное вычисление смещений в значениях между сегментами для ускорения компрессии и декомпрессии, а также упрощенная грануляция сжатых кэш-строк по конечным размерам с точностью до половины кэш-строки для лучшей адаптированности к аппаратной реализации. Как компрессия, так и декомпрессия по алгоритму ВΔI\*-HL укладываются в 1 такт при рабочей частоте 2 ГГц.

При выборе алгоритма для применения компрессии к передаваемым данным в последнее время наблюдается тенденция к рассмотрению алгоритмов на основе локальности сгруппированных значений, в то время как в более ранних работах преимущественно рассматривались алгоритмы на основе локальности изолированных значений [5]. Такой переход можно связать с тем, что при применении алгоритмов на основе локальности изолированных значений, как правило, для хранения таких значений используются справочники, которые могут формироваться как статически, так и динамически, на основе информации, передаваемой в ходе работы системы. С переходом от многоядерных процессоров на общей шине к реализации сетей-на-кристалле, сложность поддержки таких справочников существенно возрастает, т.к. придется содержать отдельные справочники для каждой пары взаимодействующих в сети устройств или даже для каждого направления передачи в этой паре. Кроме того, потребуются поддерживать консистентность справочников у приемника или передатчика. В случае использования алгоритмов на основе локальности сгруппированных значений подобных проблем не возникает, т.к. вся информация, необходимая для восстановления исходных данных, уже содержится в самих сжатых данных.

Рассмотрим теперь последние работы, посвященные разработке методик по осуществлению компрессии передаваемых данных [5 – 10].

Подход NoΔ, рассматриваемый в [5], предполагает использование VDI-подобного алгоритма в сетевых адаптерах устройств сети-на-кристалле для компрессии и декомпрессии. Рассматривая передачу флитов малой гранулярности (4 байта), авторы работы отмечают, что их подход позволяет довольно существенно снизить количество флитов, требуемое для передачи сжатых данных. Задержки на компрессию и декомпрессию при этом удается скрыть за счет того, что данные поступают на шину передачи в течение нескольких тактов. Оценка, полученная в работе, показывает, что рассматриваемый подход является более эффективным при использовании в сети-на-кристалле, чем компрессия нулевых или часто встречающихся значений.

В статье о FlitZip [6] предлагается дополнительно уменьшить число передаваемых флитов за счет частичного переноса метаданных компрессии в заголовочный флит, где часто остается неиспользованное пространство.

Авторы подхода DISCO [7, 8] выделяют дополнительную задержку на компрессию и декомпрессию, возникающую на критическом пути доступа в кэш, как одну из ключевых проблем,

сдерживающих применение аппаратной компрессии данных. В качестве решения они предлагают перенести процесс компрессии и декомпрессии данных в коммутаторы сети-на-кристалле. Возможность осуществления преобразования данных в предлагаемой ими методике определяется на основе занятости коммутаторов – компрессию или декомпрессию можно осуществить в том случае, если выдача данных в данный момент невозможна, и они находятся в буфере, ожидая разрешения на передачу. Помимо этого критерия, на решение влияет также ряд других эвристик, связанных с движением данных по сети.

Важно отметить, что все современные подходы основываются на определенной конфигурации сети – в ней должны быть сравнительно узкие каналы передачи, а также пакеты должны идти по каналам без разрывов между флитами. Требование об узких каналах передачи необходимо для того, чтобы задержка на преобразование данных была скомпенсирована задержкой на передачу одного пакета данных через определенную точку сети. Например, в случае NoD это задержка на выдачу или прием пакета в сетевом адаптере, а в случае DISCO – задержка из-за выдачи через коммутатор другого пакета, блокирующего текущий пакет, над которым осуществляется преобразование. Требование о неразрывной передаче пакетов необходимо для возможности их компрессии и декомпрессии без дополнительной задержки на ожидание всех частей пакета. Такому требованию удовлетворяют политики управления потоком store-and-forward и virtual cut-through, но для политики wormhole это требование может не соблюдаться. При работе этой политики выдача кредитов для передачи сообщений происходит на уровне флитов, а не самих пакетов. Как следствие, возможна ситуация, когда кредиты заканчиваются в момент, когда переданы еще не все флиты сообщения. Часть флитов в этом случае передается дальше, а другая остается в буфере, таким образом возникает разрыв. Авторы подхода DISCO уделяют этому внимание, предлагая компромиссный вариант с компрессией частей передаваемого пакета, состоящих из нескольких флитов, позволяющего продолжать осуществлять компрессию, хотя и с меньшей эффективностью.

Рассмотрим теперь конфигурацию сети, где используются каналы передачи большой ширины с политикой управления потоком wormhole, что не рассматривалось авторами описанных выше работ. В этом случае задержки на передачу пакета оказываются недостаточно для полноценной компенсации задержки декомпрессии, а компромиссный подход по компрессии частей пакета, предложенный

авторами DISCO, перестает работать в связи с тем, что при широких каналах передачи пакет разбивается на существенно меньшее число флитов. В связи с этим актуальной становится разработка новой методики, адаптированной для данной конфигурации сети.

**3. Цель исследования и постановка задачи.** Целью данного научного исследования является разработка методики, позволяющей применять аппаратную компрессию данных в накристалльных и межпроцессорных сетях с широкими каналами передачи данных и политикой управления потоков wormhole.

Научная новизна полученных в работе результатов определяется разработкой новой методики применения аппаратной компрессии в процессорных каналах передачи данных, предназначенной для каналов передачи большой ширины, а также разработкой ряда оптимизаций частных случаев передачи данных за счет аппаратной компрессии.

Теоретическая значимость работы заключается в том, что разработанная в ее рамках методика позволяет применять аппаратную компрессию данных в процессорных каналах передачи большой ширины, при этом как в накристалльных сетях, так и в межпроцессорных каналах связи. Практическая значимость работы заключается в возможности использования разработанной методики для повышения пропускной способности широких каналов передачи данных в процессорах общего назначения за счет аппаратной компрессии.

Общая постановка задачи исследования сводится к рассмотрению типичной конфигурации сети с широкими каналами передачи и выявлению ее особенностей по сравнению с конфигурациями, предназначенными для узких каналов, а затем к разработке методики для поддержки в такой конфигурации аппаратной компрессии, учитывающей выявленные особенности и позволяющей обрабатывать сжатые данные как в накристалльных сетях, так и в межпроцессорных каналах связи.

Эффективность применения методики оценивается на основе изменения показателей интенсивности трафика и IPC (Instructions per Clock, количества инструкций, исполняемых за процессорный такт) при применении аппаратной компрессии на основе рассматриваемой методики в модели процессора архитектуры «Эльбрус».

В качестве ограничения по применению методики рассматриваются только такие конфигурации сети, которым свойственны определенные особенности в передаче пакетов, связанные с большей шириной каналов. При этом типичный пакет с данными

в таких конфигурациях должен иметь размер, больший, чем один флит, в противном случае аппаратная компрессия не будет оказывать положительного эффекта на пропускную способность.

К допущениям при рассмотрении методики можно отнести выровненность массивов данных и их организацию в порядке байт Little Endian, а также малую задержку на компрессию и декомпрессию данных (достижимую при использовании алгоритма компрессии VDI\*-HL).

Областью применения методики является подсистема памяти современных процессоров общего назначения.

**4. Рассматриваемая конфигурация.** В качестве примера конфигурации с широкими каналами передачи и политикой управления потоком wormhole, на основе которой будет разрабатываться методика, была выбрана конфигурация сети, используемая в процессорах архитектуры «Эльбрус».

Определяющей особенностью рассматриваемой конфигурации является использование широких каналов передачи данных, выбор которых связан с необходимостью обеспечить высокий темп передачи данных процессорным ядрам. Эта характеристика напрямую связана с шириной каналов, т.к. соответствует объему данных, получаемому ядром в единицу времени. Она особенно критична из-за отсутствия в ядрах поддержки внеочередного исполнения инструкций и простаивания арифметических устройств в ожидании операндов.

Большая ширина каналов также позволяет сократить время, необходимое для выдачи или приема пакета с данными в сетевых каналах. В тактах это время соответствует числу флитов, на которые делится пакет. Передача данных одного и того же размера потребует в конфигурации с широкими каналами меньшего числа флитов, чем в конфигурации с узкими каналами.

В конфигурации сети, используемой в процессорах архитектуры «Эльбрус», типичный пакет данных, в котором передается одна кэш-строка, состоит из двух флитов, в каждом из которых передается одна из половин кэш-строки. Пакет дополняется заголовком, содержащим информацию для маршрутизации и идентификации пакета. С целью повышения пропускной способности сети для заголовков выделены отдельные линии передачи, так что в сети возможна одновременная передача одного заголовка и одного флита самого пакета.

Поскольку для заголовков в сети выделены отдельные линии, они передаются с каждым флитом пакета, т.е. с пакетом передается не один заголовок, а несколько, равное количеству флитов пакета. Подобный подход не приводит к дополнительным накладным

расходам, зато позволяет осуществлять маршрутизацию в случае разрывов между флитами одного пакета, возможных при политике управления потоком wormhole. В этом случае кредитный механизм, на основе которого работает передача между узлами сети, функционирует на уровне отдельных флитов, поэтому возможны ситуации, когда кредитов хватило для передачи только части пакета. Другой ситуацией, которая может привести к разрывам между флитами пакета, является пересинхронизация на границе доменов, регулируемых разными синхросигналами.

Особенностью передачи пакетов с данными в рассматриваемой конфигурации, по сравнению с пакетами других типов, является передача заголовков пакета на такт раньше, чем соответствующих им флитов. Это связано с тем, что за счет большей ширины каналов мультиплексирование данных начинает занимать большее время, которое сложно будет уложить в 1 такт. Передача заголовка на такт раньше позволяет заранее определить, откуда поступают данные, а затем провести мультиплексирование данных на следующем такте.

Схематичный внешний вид пакета с данными в рассматриваемой конфигурации, в котором передается одна кэш-строка, показан на рисунке 1. Пакет состоит из двух флитов, заголовки для которых передаются на такт раньше. Дополнительно вместе с заголовками передается сигнал last, обозначающий последний флит пакета.

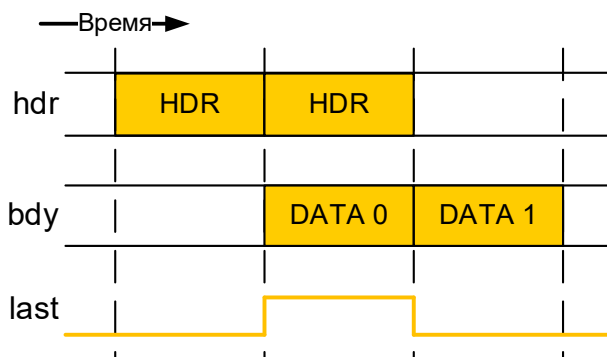


Рис. 1. Временная диаграмма передачи пакета с данными, в котором содержится одна кэш-строка

Еще одной особенностью рассматриваемой конфигурации сети является тот факт, что разбиение данных на отдельные флиты в ней производится не в сетевых адаптерах, а в самих устройствах,

до поступления данных в сеть-на-кристалле. На это есть несколько причин. Во-первых, данные для передачи могут быть готовы раньше, чем такая передача будет разрешена по протоколу когерентности. В связи с этим появляется возможность предварительно упорядочить данные в том виде, в котором они должны поступить в сеть. Во-вторых, чтение данных из памяти устройств производится, как правило, по частям, занимая несколько тактов. Поэтому, даже если данные еще не прочитаны полностью, благодаря их последовательному чтению по частям они могут быть направлены в сетевой адаптер в последовательном порядке сразу же после прочтения, по одному флиту в такт.

Межпроцессорные каналы связи подключаются к сети-на-кристалле процессора за счет контроллеров межпроцессорного обмена в количестве, равном числу процессоров, с которым будет взаимодействовать данный процессор. Пакеты, поступающие в контроллер из сети-на-кристалле, попадают в буфер, а затем производится их арбитраж для упаковки в контейнеры, являющиеся единицей передачи информации в межпроцессорных каналах связи. Данный арбитраж зависит от типа пакета (например, наивысший приоритет отдается сообщениям для завершения транзакций, а наименьший – первичным запросам, инициирующим транзакцию), их размеров и времени нахождения в буфере (чтобы, несмотря на низкий приоритет, все пакеты могли быть отправлены при высокой нагрузке на канал).

Описанные выше особенности рассматриваемой конфигурации, необходимые для работы с широкими каналами передачи, в целом слабо зависят от прочих параметров сети. Это позволяет прийти к выводу о том, что конфигурации сети, адаптированные для работы с широкими каналами передачи, которые используются в процессорах других архитектур, слабо отличаются от рассматриваемой. Таким образом, предлагаемая в работе методика может быть применена для любых конфигураций сети с широкими каналами передачи.

**5. Предлагаемая методика.** Одним из основных вопросов, которые требуется рассмотреть при интеграции аппаратной компрессии данных в процесс передачи данных, является определение точек, в которых будет производиться компрессия и декомпрессия данных. В подходах NoD и FlitZip, предложенных в [5] и [6], преобразование данных производится в сетевых адаптерах устройств, в подходе DISCO [7, 8] – в узлах сети в случаях задержки передачи данных.

При использовании рассматриваемой конфигурации сети перечисленные выше подходы не применимы. Это связано с тем, что, как в сетевых адаптерах, так и в самой сети, данные уже представлены в виде нескольких флитов. Для осуществления компрессии потребуется дождаться получения обоих флитов данных, а также поместить информацию о компрессии в заголовок пакета. С учетом задержки на компрессию в 1 такт, суммарная задержка заняла бы 3 такта, как показано на рисунке 2. Столь длительная дополнительная задержка неприемлема и не позволяет осуществлять компрессию в сетевых адаптерах или самой сети.

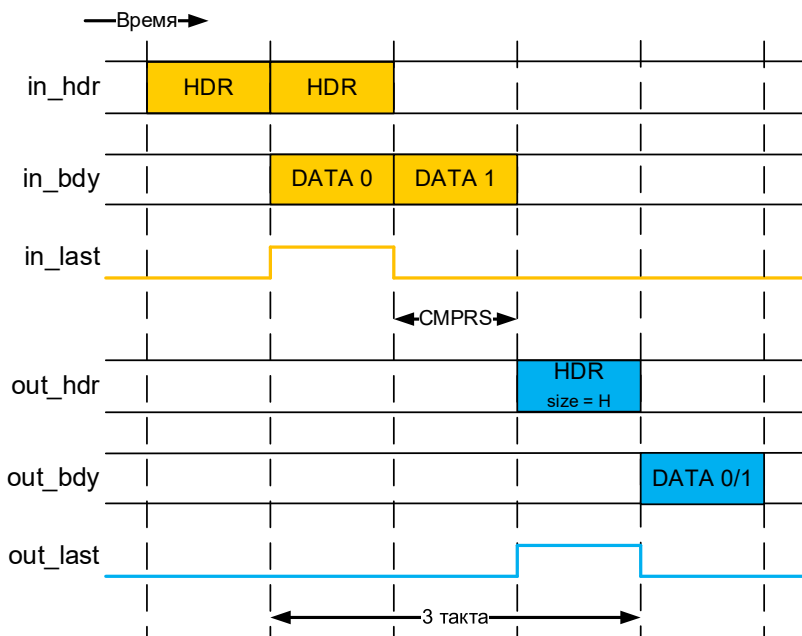


Рис. 2. Задержка при компрессии данных в сетевых адаптерах или самой сети-кристалле

При декомпрессии в сетевых адаптерах или сети дополнительной задержки, помимо задержки в 1 такт на декомпрессию, не возникнет. Во-первых, преобразование заголовка, для того чтобы убрать оттуда информацию о компрессии, можно произвести, не дожидаясь данных, а во-вторых, сжатые данные занимают только один флит, поэтому после его получения можно сразу же приступить к декомпрессии. Схематично процесс декомпрессии



представлен на рисунке 3. Единственной ситуацией, когда может потребоваться дополнительная задержка, является последовательная передача двух пакетов с данными, первый из которых является сжатым. В таком случае перед приемом второго пакета придется вставить однотактовую задержку в связи с расширением первого пакета до двух флитов в ходе декомпрессии.

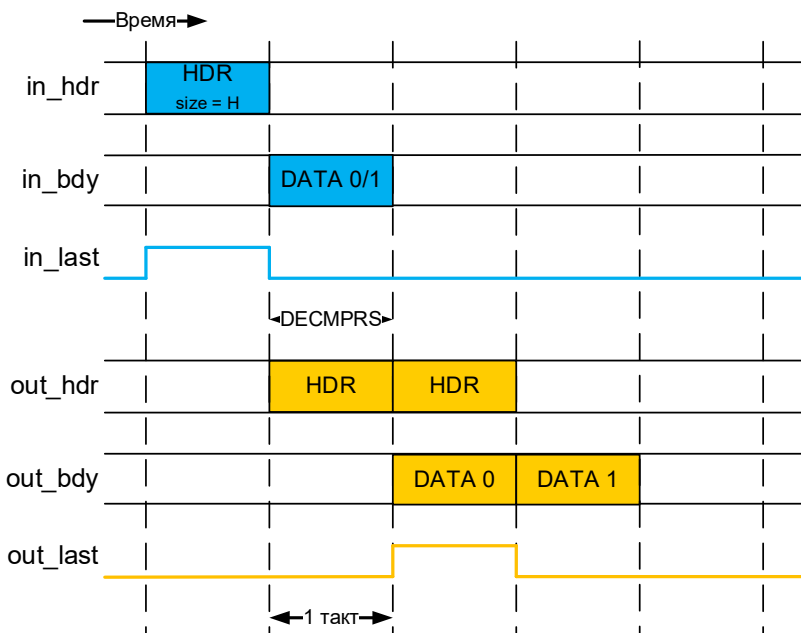


Рис. 3. Задержка при декомпрессии данных в сетевых адаптерах или самой сети-на-кристалле

Альтернативными точками применения компрессии и декомпрессии можно выбрать сами устройства, осуществляющие выдачу и прием данных. В таком случае преобразование данных будет осуществляться до передачи их в сеть, так что в сетевых адаптерах или сети данные будут передаваться в уже сжатом виде.

Такой подход особенно хорошо подходит для работы с данными записи в процессорах «Эльбрус». В протоколе когерентности, используемом в этих процессорах, записи являются непочтовыми, т.е. для отправки данных необходимо получить разрешение от получателя о готовности к приему. Интервал между готовностью данных и получением разрешения на их отправку может

использоваться для осуществления компрессии, как показано на рисунке 4. Дополнительных задержек передачи из-за компрессии при этом не возникает. При приеме данных все еще возникнет задержка на их декомпрессию. В прочих конфигурациях, где записи могут являться почтовыми, т.е. не требующими разрешения на отправку, при передаче данных будет проявляться задержка как на их компрессию, так и на декомпрессию, но, поскольку запись данных не находится на критическом пути, эта задержка не окажет существенного влияния на производительность.



Рис. 4. Компрессия данных в ходе транзакции по записи данных

При работе с данными чтения в общем случае устранить задержки невозможно, т.к., в отличие от записей, данные чтения могут быть сразу отправлены инициатору запроса, поскольку при инициировании запроса на чтение он сразу подготавливается к приему данных. Задержки могут быть скрыты частично, если разрешить устройствам, которые не модифицируют данные, таким как общая кэш-память или оперативная память, хранить данные в сжатом виде. В таком случае запись этих данных не потребует декомпрессии, а при чтении не потребуются компрессия данных, нужна будет только их декомпрессия на принимающей стороне.

Предлагаемый подход можно сочетать с методиками по компрессии данных в кэш-памяти [19] или оперативной памяти. В таком случае сжатые данные не нужно подвергать декомпрессии перед выдачей из устройства.

При передаче данных через межпроцессорные каналы связи появляется дополнительная возможность компрессии данных, связанная с их достаточно длительной буферизацией. Компрессия позволит уменьшить объем передаваемых данных минимум вдвое, а значит, появится возможность добавить в освободившееся место в контейнере другие пакеты.

Для поддержки работы со сжатыми данными в контейнерах должны поддерживаться распределения пакетов по контейнеру, где данные имеют только половинный размер. Однако в рассматриваемой конфигурации подобные пакеты уже используются – они применяются для передачи данных чтения в том случае, когда считывается только половина кэш-строки. В связи с этим переработка распределений пакетов по контейнерам в рассматриваемой конфигурации не потребуется.

**6. Дополнительные оптимизации частных случаев передачи данных.** Применяя аппаратную компрессию к сети-на-кристалле и межпроцессорным каналам связи процессора, можно дополнительно оптимизировать несколько частных случаев передачи данных.

Одним из таких случаев является передача пакетов с данными большого размера (в рассматриваемой конфигурации – вплоть до 4 кэш-строк). Такие данные могут передаваться при работе механизма Peer-to-Peer в режиме прямого доступа к памяти (Direct Memory Access, или DMA), когда два внешних по отношению к процессору устройства могут обмениваться данными напрямую без использования оперативной памяти процессора.

Использование компрессии для больших пакетов с данными может существенно сократить их размер. Как и для обычных пакетов, компрессия будет производиться на уровне одной кэш-строки, однако теперь, помимо информации, необходимой для корректной декомпрессии, вместе с каждым заголовком будет передаваться маска, показывающая, была ли определенная строка из передаваемых сжата до нулевого размера. В этом случае флиты, относящиеся к этой строке, можно не передавать. Пример подобной передачи приведен на рисунке 5, где передается большой пакет из 4-х кэш-строк. Кэш-строки 0 и 3 сжимаются до половинного размера, кэш-строка 1 не сжимается, а кэш-строка 2 сжимается до нулевого размера и не передается.

Информация о том, что кэш-строка 2 «пропущена» в передаваемых данных, отражена в передаваемой маске.

На рисунке 6 показан частный случай передачи большого пакета с данными, когда все 4 передаваемые кэш-строки нулевые. В таком случае вместо 8 флитов достаточно будет передать только 1, в котором по маске на принимающей стороне можно будет понять, что нужно восстанавливать нулевые данные.

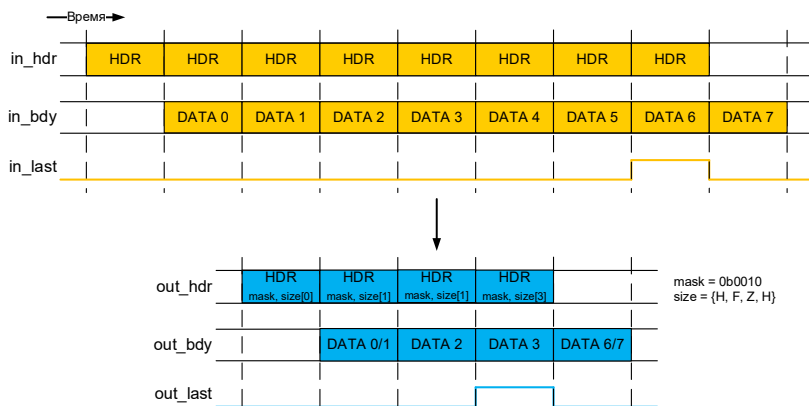


Рис. 5. Компрессия большого пакета с данными из 4 кэш-строк

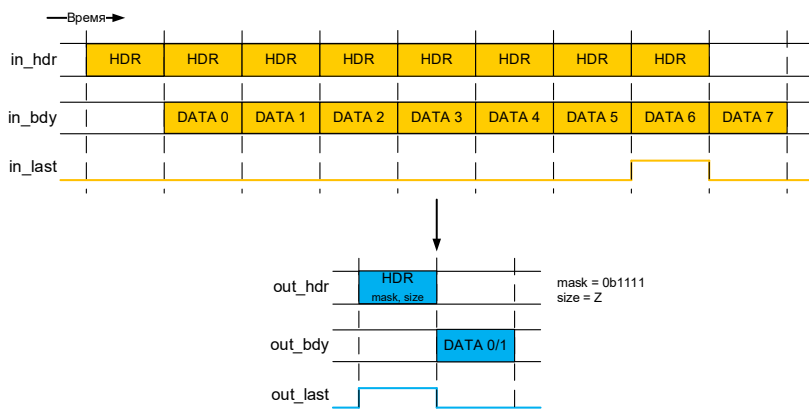


Рис. 6. Компрессия большого пакета с данными из 4 кэш-строк в случае, когда все данные нулевые

Как и при передаче пакетов стандартного размера, наиболее оптимально применение такой компрессии для данных записи, однако

в случае работы с данными чтения можно частично скрыть задержку на компрессию и декомпрессию за счет того, что число передаваемых флитов может существенно уменьшиться.

Другим частным случаем, который можно оптимизировать, является передача нулевых данных. В этом случае пакет с данными можно было бы не передавать, обходясь только передачей одного бита информации о нулевом значении данных. Для этого необходимо определить, каким именно образом эта информация может передаваться.

В случае запросов по записи информация о значении данных имеется у инициатора запроса, отправляющего данные. В связи с этим информация о нулевом значении данных может направляться сразу же вместе с первичным запросом по записи. Получатель запроса в таком случае не должен высылать подтверждение приема данных, а подготовиться к записи и самостоятельно записать нулевые значения, после чего выдать сигнал завершения транзакции. Данный порядок действий проиллюстрирован на рисунке 7.

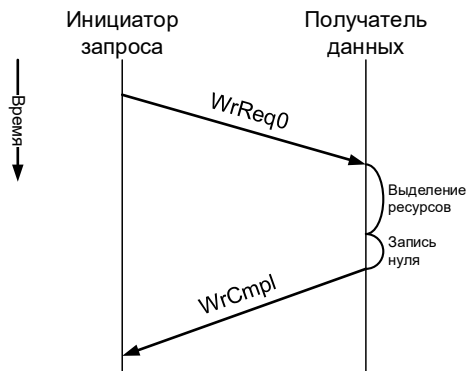


Рис. 7. Транзакция по записи нулевых данных

При запросах на чтение информация о значении данных имеется у их владельца, поэтому их потребуется передавать инициатору запроса вместе с сообщениями-ответами на первичные запросы по чтению или на снуп-запросы (запросы по чтению, поступающие от справочника устройствам, обладающим копией данных). Необходимо также учитывать, что из-за особенностей протокола когерентности устройство-владелец данных может выдавать ответы не напрямую устройству-инициатору запроса, и в таком случае для передачи информации о нулевых данных потребуется использовать

несколько сообщений-ответов. Так, на рисунке 8 показана передача ответа напрямую устройству-инициатору запроса, а на рисунке 9 взаимодействие происходит через справочник, в связи с чем ответ с признаком нулевых данных сначала отправляется также в справочник, а затем уже из справочника передается устройству-инициатору запроса.

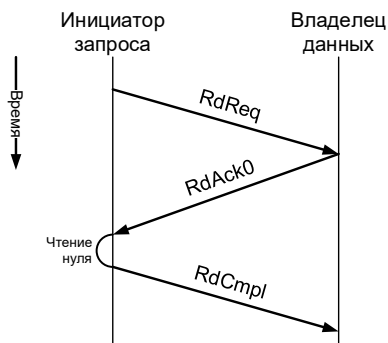


Рис. 8. Транзакция по чтению нулевых данных, передача информации о нулевом значении данных напрямую инициатору запроса

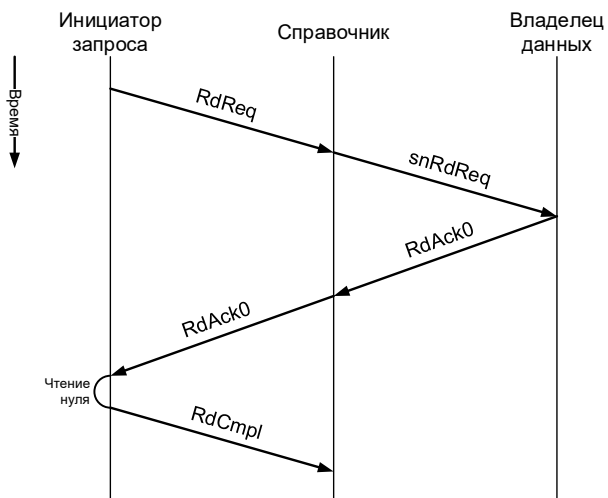


Рис. 9. Транзакция по чтению нулевых данных, передача информации о нулевом значении данных через справочник

**7. Экспериментальные результаты.** Исследование эффективности применения аппаратной компрессии при передаче данных по подсистеме памяти процессора производилось с помощью модели процессора архитектуры «Эльбрус» на основе трасс событий, изображенной на рисунке 10 и описанной в [21]. Моделируемая конфигурация соответствует процессору Эльбрус-16С и приведена в таблице 1.

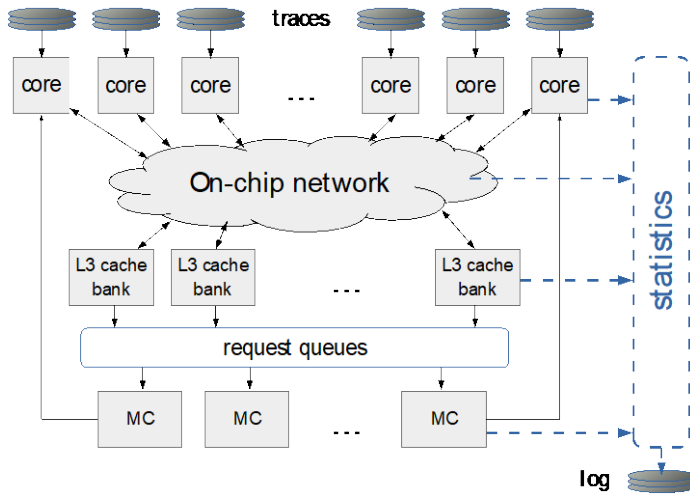


Рис. 10. Модель процессора «Эльбрус» на основе трасс событий

Таблица 1. Конфигурация процессора, используемая в симуляции

Компонент	Конфигурация
Ядро	16 VLIW-ядер, частота 2 ГГц
L1I-кэш	Частный, 128 КБ/ядро, 4-way, кэш-строка 256 Б
L1D-кэш	Частный, 64 КБ/ядро, 4-way, кэш-строка 32 Б, write-through
L2-кэш	Частный, 1 МБ/ядро, 4-way, кэш-строка 64 Б, неинклюзивный
L3-кэш	Общий, 2 МБ/ядро, 16-way, кэш-строка 64 Б, NCID
Сеть-на-кристалле	4x4 mesh, 1 запрос + 32 Б данных на один такт ядра
Оперативная память	8 каналов, DDR4 с частотой 3200 МТ/с

На рисунке 11 продемонстрирована интенсивность трафика данных в пересчете на одно ядро и один такт в сети-на-кристалле без компрессии, вызванного промахами в L2 (столбцы OCN) и L3 (столбцы

MC) для задач пакета SPEC CPU2017 при описанной в таблице 1 конфигурации процессора. Рисунок 12 показывает долю сжатых кэш-строк, достижимую при работе алгоритма компрессии в сети-на-кристалле (OCN) и межпроцессорных каналах связи (IPCC) соответственно.

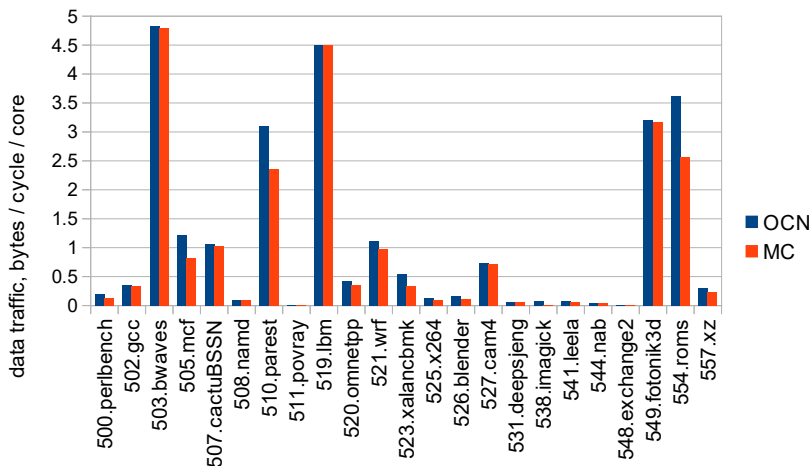


Рис. 11. Объем трафика данных сети-на-кристалле, вызванного промахами в L2 и L3

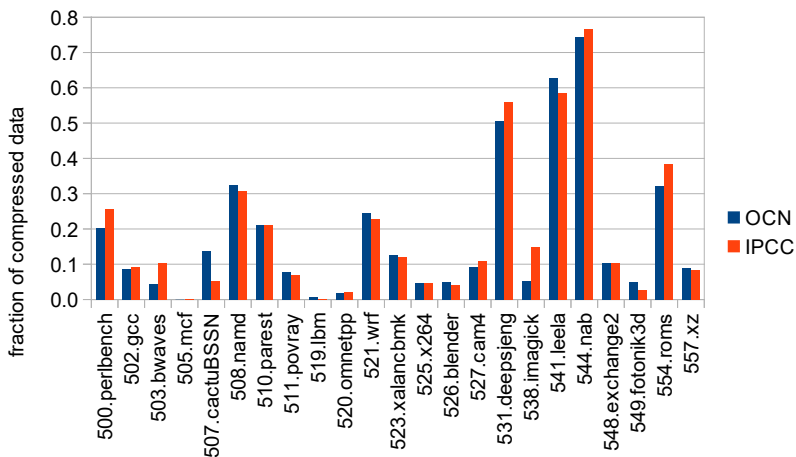


Рис. 12. Доля сжатых строк при применении алгоритма компрессии в сети-на-кристалле и межпроцессорных каналах связи



Эффективность компрессии данных в сети-на-кристалле моделировалась с ограничением пропускной способности этой сети по данным до 4 байт за такт на ядро, что соответствует одному 32-байтному флиту за 8 тактов для случаев большого количества ядер, другой топологии или/и меньшей частоты сети относительно ядер. Относительное увеличение IPC (Instructions per Clock, количество инструкций, исполняемых за процессорный такт) за счет использования аппаратной компрессии данных в тестах пакета SPEC CPU2017 при такой конфигурации, а также среднее геометрическое этого значения по всем задачам, представлено на рисунке 13. Наибольший прирост (вплоть до 7,1%) наблюдается в задачах 554.roms, 503.bwaves и 507.cactuBSSN, где в сети одновременно наблюдается высокий трафик данных (из-за частых промахов в L2 кэш) и высокая доля сжатых данных. В процессоре с меньшим размером L2 кэша или менее оптимизированных задачах эффект от аппаратной компрессии данных может быть ещё больше.

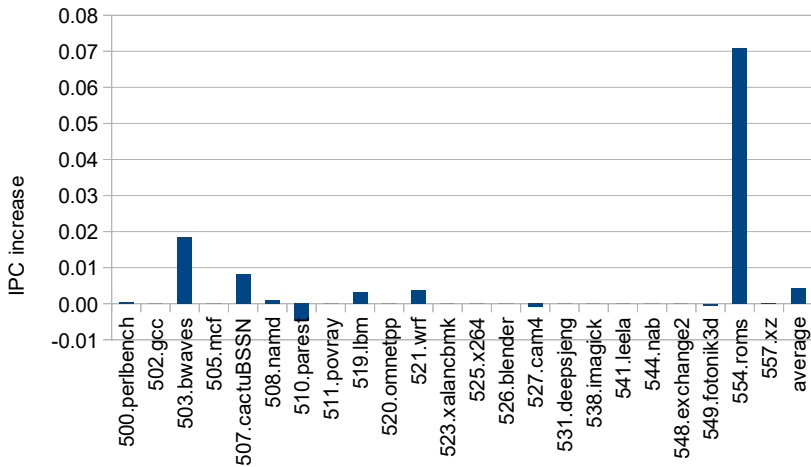


Рис. 13. Относительное увеличение IPC для задач пакета SPEC CPU2017 при аппаратной компрессии в сети-на-кристалле

Для моделирования применения аппаратной компрессии данных в межпроцессорных каналах связи пропускная способность доступа к некоторым страницам памяти (при их размере в 2 МБ) была ограничена величиной 10 ГБ/с с добавлением задержки в 200 нс. Передача сжатого пакета данных при этом составляла 70 байт, а несжатого – 103 байт суммарного трафика по каналу. Относительное

увеличение IPC благодаря компрессии в зависимости от доли таких страниц на такой модели показано на рисунке 14. Приведены результаты, полученные на задачах пакета SPEC CPU2017, а также среднее геометрическое по всем задачам.

С увеличением доли страниц, располагающихся в чужом процессоре, увеличивается среднее время доступа в память, из-за чего в некоторых задачах уменьшается трафик; соответственно, эффект от оптимизации зависит от доли таких страниц по-разному для разных задач. В среднем наибольший относительный прирост IPC при использовании компрессии наблюдается при 50% страниц в памяти чужого процессора, составляя 1,4%. Максимальный прирост, равный 14,0%, наблюдается в тесте 554.roms при 25% страниц в памяти чужого процессора.

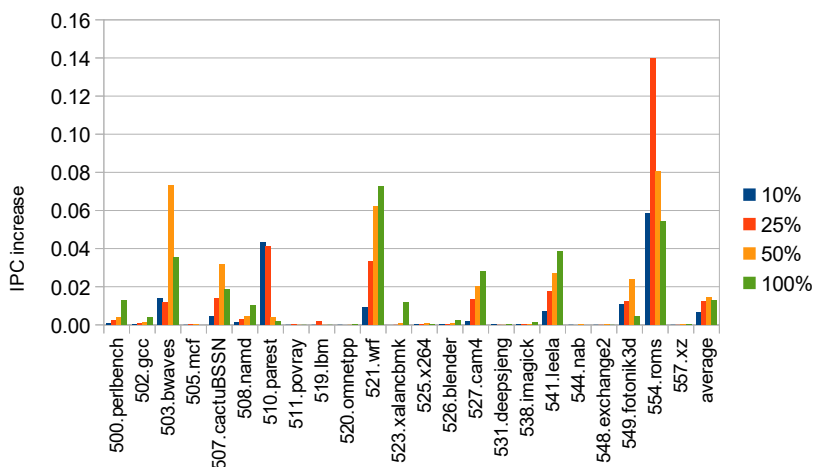


Рис. 14. Относительное увеличение IPC для задач пакета SPEC CPU2017 с аппаратной компрессией в межпроцессорных каналах связи при различных долях страниц памяти в чужих процессорах

**8. Выводы.** В рамках данной работы была достигнута цель научного исследования, заключающаяся в разработке новой методики применения аппаратной компрессии передаваемых данных в подсистеме памяти процессоров общего назначения, позволяющей использовать ее во внутрипроцессорных и межпроцессорных сетях с широкими каналами передачи данных и политикой управления потоком wormhole.

Применение новой методики в сети-на-кристалле предполагает перенос компрессии и декомпрессии данных в устройства сети. За счет этого задержки на преобразование данных в ходе компрессии или декомпрессии могут быть частично скрыты для запросов по чтению данных и полностью скрыты для запросов по записи данных.

В межпроцессорных каналах связи в рамках новой методики компрессию данных предлагается осуществлять без задержек, пользуясь фактом буферизации данных в контроллере межпроцессорных каналов перед их отправкой.

Дополнительно в рамках исследования разрабатываются оптимизации передачи пакетов с данными большого размера и передачи нулевых данных. Передача пакетов большого размера оптимизируется за счет передачи в заголовке маски нулевых кэш-строк, позволяющей не отправлять флиты большого пакета, относящиеся к нулевым кэш-строкам. Передачу нулевых данных предлагается оптимизировать за счет того, что вместо пакета с нулевыми данными вместе с другими сообщениями протокола когерентности будет передаваться признак нулевых данных.

Эффективность рассматриваемой методики подтверждается результатами экспериментов, в рамках которых оценивается применение аппаратной компрессии данных на основе методики к сети-на-кристалле и межпроцессорным каналам связи процессора в ходе работы задач пакета SPEC CPU2017.

В сети-на-кристалле, модифицированной для повышения нагрузки на нее, на некоторых задачах SPEC CPU2017 наблюдается относительное увеличение IPC до 7,1%. Однако при этом на сравнительно большом количестве задач относительного прироста IPC не наблюдается, что, согласно проведенным измерениям, можно связать либо с малым объемом трафика в сети-на-кристалле в этих задачах (т.е. с малым числом промахов в L2-кэш), либо с плохой сжимаемостью данных в задаче. Анализируя полученные результаты, можно сделать вывод о том, что задачи, в которых одновременно будет достаточно большой трафик в сети и высокая доля сжатых строк, получили бы наибольшее ускорение за счет аппаратной компрессии.

При моделировании использования аппаратной компрессии в межпроцессорных каналах связи было достигнуто относительное увеличение IPC вплоть до 14,0%. В целом максимальная степень относительного увеличения IPC для конкретных задач достигается при разных долях страниц в других процессорах, что показывает, что в определенных случаях положительный эффект от применения

компрессии может ограничиваться общей задержкой на передачу данных по межпроцессорным каналам связи.

Результаты работы позволяют применять аппаратную компрессию данных в сетях с широкими каналами передачи и политикой управления потоком wormhole, а также подтверждают, что аппаратная компрессия данных является достаточно эффективным механизмом повышения производительности подсистемы памяти.

### Литература

1. Serpa M.S., Moreira F.B., Navaux P.O., Cruz E.H., Diener M., Griebler D., Fernandes L.G. Memory performance and bottlenecks in multicore and GPU architectures. 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2019. pp. 233–236.
2. Mohamed A.M., Mubark N., Zagloul S. Performance aware shared memory hierarchy model for multicore processors. Scientific Reports. 2023. vol. 13(1). no. 7313.
3. Iyer R., De V., Illikkal, R., Koufaty, D., Chitlur, B., Herdrich, A., Khellah M., Hamzaoglu F., Karl E. Advances in microprocessor cache architectures over the last 25 years. IEEE Micro. 2021. T. 41. № 6. C. 78–88.
4. Papazian I.E. New 3rd Gen Intel® Xeon® Scalable Processor (Codename: Ice Lake-SP) // Hot Chips Symposium. 2020. C. 1–22.
5. Zhan J., Poremba M., Xu Y., Xie Y. No $\Delta$ : Leveraging delta compression for end-to-end memory access in NoC based multicores. 19th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2014. pp. 586–591.
6. Deb D., Rohith M.K., Jose J. Flitzip: Effective packet compression for noc in multiprocessor system-on-chip // IEEE Transactions on Parallel and Distributed Systems. 2021. T. 33. № 1. pp. 117–128.
7. Wang Y., Han Y., Zhou J., Li H., Li X. DISCO: A low overhead in-network data compressor for energy-efficient chip multi-processors // Proceedings of the 53rd Annual Design Automation Conference. 2016. C. 1–6.
8. Wang Y., Li H., Han Y., Li X. A low overhead in-network data compressor for the memory hierarchy of chip multiprocessors // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2017. vol. 37. no. 6. pp. 1265–1277.
9. Li X., Sondhi T. FlitReduce: Improving Memory Fabric Performance via End-to-End Network Packet Compression. UC Berkeley CS262A Report. 2021. 9 p.
10. Pullaiah T., Manjunathachari K., Malleswari B.L. B $\Delta$ -NIS: Performance analysis of an efficient data compression technique for on-chip communication network. Integration. 2023. vol. 89. pp. 83–93.
11. Pekhimenko G., Seshadri V., Mutlu O., Gibbons P.B., Kozuch M.A., Mowry T.C. Base-delta-immediate compression: Practical data compression for on-chip caches // Proceedings of the 21st international conference on Parallel architectures and compilation techniques. 2012. C. 377–388.
12. Gaur J., Alameldeen A.R., Subramoney S. Base-victim compression: An opportunistic cache compression architecture // ACM SIGARCH Computer Architecture News. 2016. vol. 44. no. 3. pp. 317–328.
13. Carvalho D.R., Sez nec A. Understanding cache compression // ACM Transactions on Architecture and Code Optimization (TACO). 2021. vol. 18. no. 3. pp. 1–27.
14. Pekhimenko G., Seshadri V., Kim Y., Xin H., Mutlu O., Gibbons P.B., Kozuch M.A., Mowry T.C. Linearly compressed pages: A low-complexity, low-latency main memory compression framework // Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture. 2013. C. 172–184.

15. Young V., Kariyappa S., Qureshi M.K. CRAM: Efficient Hardware-Based Memory Compression for Bandwidth Enhancement // arXiv preprint arXiv:1807.07685. 2018.
16. Choukse E., Erez M., Alameldeen A.R. Compresso: Pragmatic main memory compression // 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2018. С. 546–558.
17. Сурченко А.В. Исследование применимости аппаратной компрессии данных в межпроцессорных каналах связи процессоров с архитектурой Эльбрус // Труды Института системного программирования РАН. 2022. Т. 34. № 1. С. 49–58.
18. Thuresson M., Spracklen L., Stenstrom P. Memory-link compression schemes: A value locality perspective // IEEE Transactions on Computers. 2008. vol. 57. no. 7. pp. 916–927.
19. Kozhin A.S., Surchenko A.V. Design of Data Compression Mechanism in Cache Memory of Elbrus Processors // International Conference Engineering and Telecommunication (En&T). IEEE, 2020. С. 1–5.
20. Nedbailo Y.A., Surchenko A.V., Bychkov I.N. Reducing miss rate in a non-inclusive cache with inclusive directory of a chip multiprocessor // Computer Research and Modeling. 2023. vol. 15. no. 3. pp. 639–656.
21. Nedbailo Y. Fast and scalable simulation framework for large in-order chip multiprocessors // 26th Conference of Open Innovations Association (FRUCT). IEEE, 2020. pp. 335–345.

**Сурченко Александр Викторович** — старший инженер, АО "МЦСТ"; преподаватель кафедры, кафедра информатики и вычислительной техники, МФТИ (НИУ). Область научных интересов: производительность и поддержка когерентности подсистемы памяти процессоров общего назначения, аппаратная компрессия данных, кэш-память. Число научных публикаций — 10. Alexander.V.Surchenko@mcst.ru; улица Вавилова, 24, 119049, Москва, Россия; р.т.: +7(499)135-8969.

**Недбайло Юрий Александрович** — канд. техн. наук, ведущий инженер, АО "МЦСТ"; ведущий инженер, ПАО «ИНЭУМ им. И.С. Брука». Область научных интересов: подсистема памяти процессоров общего назначения. Число научных публикаций — 25. yuri.nedbailo@mail.ru; улица Вавилова, 24, 119049, Москва, Россия; р.т.: +7(916)936-8670.

A. SURCHENKO, YU. NEDBAILO  
**HARDWARE COMPRESSION METHOD FOR ON-CHIP AND  
INTERPROCESSOR NETWORKS WITH WIDE CHANNELS AND  
WORMHOLE FLOW CONTROL POLICY**

*Surchenko A., Nedbailo Yu.* **Hardware Compression Method for On-Chip and Interprocessor Networks with Wide Channels and Wormhole Flow Control Policy.**

**Abstract.** Increasing the number of processing cores is currently a common way to boost processor performance. However, the load on the memory subsystem consequently increases as the number of its agents grows. Hardware data compression is an unconventional approach to improving memory subsystem performance by reducing, firstly, the main memory access rate by increasing the cache capacity and, secondly, data traffic by packing the data more densely. The paper describes the implementation of hardware data compression in the on-chip network and interprocessor links of a configuration with wide data transmission channels and a wormhole flow control policy. The existing solutions cannot be applied to such configurations because they are essentially based on using narrow data channels and flow control policies implying uninterrupted packet transmission, which is not maintained with the wormhole flow control. The method proposed in this paper enables the use of hardware compression in the aforementioned configuration by moving data compression and decompression from networks to the connected devices, as well as by using a number of optimizations to hide the data processing delays. Optimizations of some specific cases, such as the transmission of large data packets with several cache lines or the transmission of zero data, are considered. Special attention is given to data transmission via interprocessor links, where, due to their lower bandwidth compared to the on-chip network, data compression can be the most beneficial. The increase in memory subsystem bandwidth from using hardware data compression was confirmed in the experiments showing the relative IPC increase in SPEC CPU2017 benchmarks up to 14 percent.

**Keywords:** processor architecture, memory subsystem, hardware data compression, network-on-chip, interprocessor links, processor model.

## References

1. Serpa M.S., Moreira F.B., Navaux P.O., Cruz E.H., Diener M., Griebler D., Fernandes L.G. Memory performance and bottlenecks in multicore and GPU architectures. 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2019. pp. 233–236.
2. Mohamed A.M., Mubark N., Zagloul S. Performance aware shared memory hierarchy model for multicore processors. Scientific Reports. 2023. vol. 13(1). no. 7313.
3. Iyer R., De V., Illikkal, R., Koufaty, D., Chitlur, B., Herdrich, A., Khellah M., Hamzaoglu F., Karl E. Advances in microprocessor cache architectures over the last 25 years. IEEE Micro. 2021. T. 41. № 6. C. 78–88.
4. Papazian I.E. New 3rd Gen Intel® Xeon® Scalable Processor (Codename: Ice Lake-SP). Hot Chips Symposium. 2020. C. 1–22.
5. Zhan J., Poremba M., Xu Y., Xie Y. No $\Delta$ : Leveraging delta compression for end-to-end memory access in NoC based multicores. 19th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2014. pp. 586–591.
6. Deb D., Rohith M.K., Jose J. Flitzip: Effective packet compression for noc in multiprocessor system-on-chip. IEEE Transactions on Parallel and Distributed Systems. 2021. T. 33. № 1. pp. 117–128.

7. Wang Y., Han Y., Zhou J., Li H., Li X. DISCO: A low overhead in-network data compressor for energy-efficient chip multi-processors. Proceedings of the 53rd Annual Design Automation Conference. 2016. C. 1–6.
8. Wang Y., Li H., Han Y., Li X. A low overhead in-network data compressor for the memory hierarchy of chip multiprocessors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2017. vol. 37. no. 6. pp. 1265–1277.
9. Li X., Sondhi T. FlitReduce: Improving Memory Fabric Performance via End-to-End Network Packet Compression. UC Berkeley CS262A Report. 2021. 9 p.
10. Pullaiah T., Manjunathachari K., Malleswari B.L.  $\Delta$ -NIS: Performance analysis of an efficient data compression technique for on-chip communication network. Integration. 2023. vol. 89. pp. 83–93.
11. Pekhimenko G., Seshadri V., Mutlu O., Gibbons P.B., Kozuch M.A., Mowry T.C. Base-delta-immediate compression: Practical data compression for on-chip caches. Proceedings of the 21st international conference on Parallel architectures and compilation techniques. 2012. C. 377–388.
12. Gaur J., Alameldeen A.R., Subramoney S. Base-victim compression: An opportunistic cache compression architecture. ACM SIGARCH Computer Architecture News. 2016. vol. 44. no. 3. pp. 317–328.
13. Carvalho D.R., Sez nec A. Understanding cache compression. ACM Transactions on Architecture and Code Optimization (TACO). 2021. vol. 18. no. 3. pp. 1–27.
14. Pekhimenko G., Seshadri V., Kim Y., Xin H., Mutlu O., Gibbons P.B., Kozuch M.A., Mowry T.C. Linearly compressed pages: A low-complexity, low-latency main memory compression framework. Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture. 2013. C. 172–184.
15. Young V., Kariyappa S., Qureshi M.K. CRAM: Efficient Hardware-Based Memory Compression for Bandwidth Enhancement. arXiv preprint arXiv:1807.07685. 2018.
16. Choukse E., Erez M., Alameldeen A.R. Compresso: Pragmatic main memory compression. 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2018. C. 546–558.
17. Surchenko A.V. [Evaluation of hardware data compression in interprocessor links of Elbrus processors]. Trudy Instituta sistemnogo programirovaniya RAN – Proceedings of the Institute for System Programming of the RAS. 2022. vol. 34. no. 1. pp. 49–58.
18. Thuresson M., Spracklen L., Stenstrom P. Memory-link compression schemes: A value locality perspective. IEEE Transactions on Computers. 2008. vol. 57. no. 7. pp. 916–927.
19. Kozhin A.S., Surchenko A.V. Design of Data Compression Mechanism in Cache Memory of Elbrus Processors. International Conference Engineering and Telecommunication (En&T). IEEE, 2020. C. 1–5.
20. Nedbailo Y.A., Surchenko A.V., Bychkov I.N. Reducing miss rate in a non-inclusive cache with inclusive directory of a chip multiprocessor. Computer Research and Modeling. 2023. vol. 15. no. 3. pp. 639–656.
21. Nedbailo Y. Fast and scalable simulation framework for large in-order chip multiprocessors. 26th Conference of Open Innovations Association (FRUCT). IEEE, 2020. pp. 335–345.

**Surchenko Alexander** — Senior engineer, JSC "MCST"; Lecturer of the department, Department of computer science and computer engineering, MIPT. Research interests: performance and coherence support of memory subsystem of general-purpose processors, hardware data compression, cache memory. The number of publications — 10. Alexander.V.Surchenko@mcst.ru; 24, Vavilova St., 119049, Moscow, Russia; office phone: +7(499)135-8969.

**Nedbailo Yuri** — Ph.D., Lead engineer, JSC "MCST"; Lead engineer, JSC "INEUM named after I.S. Bruk". Research interests: memory subsystem of general-purpose processors. The number of publications — 25. [yuri.nedbailo@mail.ru](mailto:yuri.nedbailo@mail.ru); 24, Vavilova St., 119049, Moscow, Russia; office phone: +7(916)936-8670.



Ф.В. ГУНДЕЛАХ, Л.А. СТАНКЕВИЧ  
**КЛАССИФИКАЦИЯ ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ  
ПАТТЕРНОВ НА ОСНОВЕ НЕЙРОМОРФНЫХ СЕТЕЙ**

*Гунделах Ф.В., Станкевич Л.А. Классификация пространственно-временных паттернов на основе нейроморфных сетей.*

**Аннотация.** Эта работа посвящена проблемам разработки нейроморфных классификаторов пространственно-временных паттернов, а также их применению в нейроинтерфейсах для решения задачи управления робототехническими устройствами. Рассматриваются классификаторы пространственно-временных паттернов на основе нейронных сетей, метода опорных векторов, глубоких нейронных сетей, римановой геометрии. Проводится сравнительное исследование этих классификаторов на точность многоклассового распознавания электроэнцефалографических сигналов, показывающих зависимость от времени биоэлектрическую активность в различных зонах мозга при воображении разных движений. Показано, что такие классификаторы могут обеспечить точность 60-80% при распознавании от двух до четырех классов воображаемых движений. Предложен новый тип классификатора на основе нейроморфной сети, биоподобные нейроны которой построены на модели Ижикевича. Исходный электроэнцефалографический сигнал кодируется в импульсные потоки на основе алгоритма временного кодирования. Предложенная нейроморфная сеть обрабатывает импульсные входные последовательности и формирует на выходах импульсные потоки разной частоты. Обучение сети проводится по размеченной информации, содержащей примеры правильного распознавания нужных классов паттернов воображаемых движений с применением алгоритма Supervised STDP. Распознанный класс паттерна воображаемого движения определяется по максимальной частоте импульсного потока выходной последовательности. Нейроморфный классификатор показал среднюю точность классификации 90% для 4-х классов воображаемых двигательных команд, а максимальная точность составила 95%. Путем моделирования задачи управления роботом в виртуальной среде показано, что такая точность классификации достаточна для эффективного применения классификатора в составе неинвазивного интерфейса «мозг-компьютер» при бесконтактном управлении робототехническими устройствами.

**Ключевые слова:** пространственно-временные паттерны, классификация, нейронные сети, опорные вектора, риманова геометрия, нейроморфные сети, нейроинтерфейс, электроэнцефалография, воображаемые движения, бесконтактное управление.

**1. Введение.** В настоящее время при контроле сложных процессов, а также диагностике и управлении сложными устройствами часто требуется решать задачи комплексного распознавания сигналов, имеющих определенную продолжительность во времени и снятых в разных точках пространства. Совокупность таких сигналов можно определить, как пространственно-временные паттерны (ПВП). Классификация таких паттернов позволяет определить ситуации процессов или действия объектов.

Одним из примеров ПВП является набор сигналов, полученных в определенном временном интервале от нескольких пространственно

разделенных датчиков, фиксирующих разные параметры технологического процесса [1, 2]. Классификация таких паттернов позволяет осуществлять контроль правильности протекания процесса. Другим примером является видеонаблюдение за несколькими объектами, траектории движения которых во времени вычисляются с помощью системы технического зрения, следящей за этими объектами [3, 4]. Классификация ПВП, составленных из таких сигналов, позволяет выявить ситуации, соответствующие опасным действиям наблюдаемых объектов.

В последнее время большое внимание уделяется разработке различных вариантов нейроинтерфейсов, в которых требуется распознавать сигналы биоэлектрической активности мозга или информации, передаваемой по нервам, в пространственно разделенных зонах мозга и периферийной нервной системы и определенных временных интервалах. Классификация таких ПВП дает возможность диагностировать состояния мозга, распознавать передаваемые по нервам команды и даже выявлять воображаемые человеком действия [5, 6]. Одним из распространенных применений нейроинтерфейсов является их использование для бесконтактного управления робототехническими устройствами, которые могут обеспечить лучшее качество жизни людям с ограниченными возможностями или расширить возможности здоровых людей. Так, обездвиженные люди могут через нейроинтерфейсы управлять электрическими инвалидными креслами, специализированными мобильными платформами, протезами верхних и нижних конечностей, манипуляторами типа третьей руки, экзоскелетами нижних и верхних конечностей, а также тела в целом [7, 8, 9]. Здоровые люди также могут использовать нейроинтерфейсы, например, для управления или роботизированными бытовыми устройствами, такими как пылесосы или стиральные машины, а также персонажами компьютерных игр и тренажеров [10]. Предполагается, что в ближайшем будущем нейроинтерфейсы могут быть использованы для бесконтактного управления аватарами, например, в виртуальной реальности [11] или виде автономных антропоморфных роботов, которые, в 21-м веке могут стать полноценными помощниками человека в быту и на производстве.

Классификация ПВП разного типа включает, как правило, этапы регистрации и подготовки набора входящих в них сигналов, а также использование различных методов вычисления степени принадлежности паттернов к определенным классам, на которые настроен классификатор. Каждый из этапов имеет свои особенности.

Так, для регистрации параметров технологических процессов используются набор отдельных датчиков, а параметры траекторий наблюдаемых объектов вычисляются по изображениям, получаемым с помощью видеокамеры. Такие средства регистрации используются давно, хорошо изучены и относительно просты в применении. В нейроинтерфейсах чаще всего в качестве средств получения информации о биоэлектрической активности мозга используются приборы, регистрирующие электроэнцефалографические (ЭЭГ) сигналы, а с нервов снимаются электромиографические (ЭМГ) сигналы. Популярность этих средств обусловлена тем, что при регистрации ЭЭГ или ЭМГ не требуется хирургическое вмешательство, т.е. средства регистрации являются неинвазивными и, как правило, недорогими.

Этап вычисления степени принадлежности ПВП к определенному классу из заданного набора классов также имеет свои особенности. ПВП, содержащие наборы параметров технологических процессов, классифицируются относительно просто и точно, например, с использованием метода опорных векторов, поскольку состояния процессов, как правило, хорошо различимы. Более сложно и с меньшей точностью можно классифицировать ПВП, полученные от системы технического зрения при видеонаблюдении, поскольку вычисление параметров движений объектов может производиться с ошибками. При этом наилучшие результаты по точности, как правило, могут дать классификаторы на основе нейронных сетей. Наиболее сложно классифицировать ПВП в нейроинтерфейсах, поскольку сигналы ЭЭГ и ЭМГ вариативны и сильно зашумлены. Для классификации таких паттернов могут использоваться различные методы, среди которых можно выделить методы, основанные на методе опорных векторов, нейронных сетях с глубоким обучением, римановой геометрии и различных комбинациях этих методов.

В последнее время начали разрабатывать и применять классификаторы, основанные на нейроморфных сетях, компоненты которых реализованы на биоподобных моделях нейронов. Предполагается, что такие нейроморфные классификаторы могут обеспечить высокую точность классификации ПВП и могут быть использованы в нейроинтерфейсах в системах управления устройствами в реальном времени и при большой вариативности и зашумленности сигналов.

Целью этой работы является разработка классификатора ПВП на основе нейроморфных сетей и оценка эффективности его

применения в нейроинтерфейсах для бесконтактного управления роботизированными устройствами.

Далее в секции 2 этой работы рассматриваются классификаторы пространственно-временных паттернов на основе нейронных сетей, опорных векторов, глубоких нейронных сетей, римановой геометрии. Проводится сравнительное исследование этих классификаторов на точность многоклассового распознавания электроэнцефалографических сигналов, показывающих зависимость от времени биоэлектрическую активность в различных зонах мозга при воображении разных движений. Показано, что такие классификаторы пока не могут обеспечить достаточную для бесконтактного управления точность для четырех классов воображаемых движений. В секции 3 описан специально разработанный классификатор повышенной точности на нейроморфной сети, биоподобные нейроны которой построены на модели Ижикевича. В секции 4 на основе моделирования показано, что достигнутая точность достаточна для эффективного применения классификатора в составе нейроинтерфейсов при бесконтактном управлении робототехническими устройствами. Секция 5 является заключительной, где подведены итоги исследования, даны некоторые рекомендации по разработке и применению нейроморфных классификаторов и намечены будущие исследования по их совершенствованию.

## **2. Классификаторы пространственно-временных паттернов.**

Существующие классификаторы ПВП используют разные принципы обработки исходной информации. В первых вариантах классификаторов каждая последовательность регистрируемых отсчетов в определенном интервале времени подвергалась фильтрации для уменьшения уровня шума и удаления артефактов, после чего по специальным алгоритмам вычислялись векторы признаков, отображающие временные и пространственные особенности паттерна. При этом размерности векторов признаков были значительно меньше размерности исходной последовательности. По вычисленному набору векторов признаков и производилась собственно классификация ПВП. Так обычно делается в классификаторах на основе метода опорных векторов (МОВ) или искусственных нейронных сетей (ИНС) типа многослойных персептронов. В последних вариантах фаза вычисления специальных признаков была исключена, и классификацию стали проводить сразу после фильтрации и удаления артефактов. Это стало возможным в классификаторах на сверточных нейронных сетях (СНС), римановой геометрии и спайковых нейронных сетях (СпНС). В таких классификаторах происходит автоматическое снижение размерности

паттерна и перевод его в матрицу весов (для СНС), набор коэффициентов ковариации (для римановой геометрии) или импульсные последовательности (для СпНС). В настоящее время для классификации ПВП используются разные методы, такие как опорные вектора, нейронные сети типа персептронов, сверточные нейронные сети, риманова геометрия и нейроморфные сети.

В этой секции представлены результаты сравнительного исследования разных методов классификации с позиций точности классификации ПВП и возможности принятия решений в реальном времени.

Первым вариантом был двухуровневый комитет классификаторов, основанный на ИНС типа многослойного персептрона и МОВ [12, 13]. Первый уровень состоял из двух локальных классификаторов на ИНС типа многослойного персептрона с двумя скрытыми слоями и выходным слоем (в скрытых слоях использовалась сигмоидальная функция, а в выходном слое – линейная функция), а также и двух классификаторов на основе МОВ с ядром в виде радиально-базисной функции. Второй уровень представлял собой ИНС для объединения результатов классификаторов первого уровня. Классификаторы первого уровня анализировали отдельные векторы признаков: длина под кривой и кепстральные коэффициенты, полученные для 20 сегментов входного сигнала (всего 40 признаков) и выдавали ответы принадлежности сигнала пробы к тому или иному классу (4 класса). На основе этих сигналов формировался вектор признаков (16 компонентов) для ИНС второго уровня, который принимал окончательное решение о принадлежности данного ПВП к определенному классу из числа обученных.

Второй вариант классификатора был построен на базе специализированной СНС с глубоким обучением. В нашем случае для исследований была выбрана архитектура нейронной сети типа *Shallow ConvNet* с 4-мя слоями, способная классифицировать сырые сигналы ЭЭГ [14]. СНС включала: (1) слой свертки во времени (*Temporal Convolution Layer*): форма входного сигнала 21x300; размер ядра – 1x25; используется 40 фильтров; окно в 25 отсчетов выбрано исходя из формы сигнала ЭЭГ; (2) слой свертки в пространстве (*Spatial Convolution Layer*): размер входного тензора – 40 (количество фильтров) x 21 (количество каналов) x 276 (количество отсчетов после свёртки по времени), размер ядра – 40x21; (3) усреднение в окне (*Mean Pooling*): размер входной матрицы – 40x276, размера ядра – 1x75; (4) решающий слой (*softmax*): размер входной матрицы после *Mean Pooling* 40x14. В качестве функции активации в скрытых и выходном

слоях использована экспоненциальная линейная функция (Exponential Linear Unit – ELU).

Третьим вариантом был специально разработанный классификатор на основе римановой геометрии, который, как и классификатор на СНС, не требует предварительного вычисления признаков, т.к. они вычисляются автоматически внутри классификатора [15]. Классификатор на римановой геометрии использует специфичное определение расстояния для сравнения ковариационных матриц, благодаря которому матрицы проб, относящихся к одному классу, оказываются ближе друг к другу, нежели матрицы проб из разных классов. Для реализации классификатора на основе римановой геометрии в данной работе использован алгоритм Minimum Distance to Mean, который является обобщением метода ближайших соседей для случая ковариационных матриц. Этот алгоритм вычисляет среднюю точку в пространстве признаков для каждого из классов. Данный подход интересен тем, что он работает одинаково для любого количества классов и имеет хорошую способность к обобщению при значительной вариативности исходных данных.

Четвертый вариант – классификатор ПВП, реализованный на СпНС [16]. Заметим, что использование СпНС для классификации временных последовательностей является относительно новым. В описанном классификаторе использована двухслойная СпНС, построенная на импульсных нейронах на основе самой простой модели Интегрирования и поджигания (Leaky-Integrate-and-Fire – LIF), включающей одно дифференциальное уравнение.

При исследовании этих классификаторов распознавались ПВП, содержащие наборы сигналов ЭЭГ, соответствующие различным воображаемым движениям. Тестирование проводилось на 10-ти испытуемых сначала с первыми 3-мя вариантами описанных классификаторов.

При тестировании комитета классификаторов в автономных сеансах при классификации паттернов ЭЭГ воображаемых команд без накопления проб вероятность успешной классификации 2-х типов воображаемых команд (сжатие левой и правой кисти) для испытуемых, обучившимся использованию нейроинтерфейса, точность классификации в среднем составляла  $63 \pm 9\%$ . Если считать только сессии, выполненные после овладения испытуемым навыком обращения с нейроинтерфейсом, то средняя точность составляла  $70 \pm 10\%$ . В процессе тестирования системы максимальная точность классификации составила 93% (таблица 1). В то же время

исследования показали, что небольшая часть испытуемых (около 30%) не способна научиться эффективно работать с нейроинтерфейсом на основе сигналов ЭЭГ (таблица 1). Кроме того, исследования показали, что характеристики классификатора соответствуют требованиям реального времени. Временные задержки, необходимые для получения ответа от классификатора, находились в пределах 150 мс, что является приемлемым для использования нейроинтерфейсов с биологически обратной связью.

При тестировании классификатора на основе СНС на его вход подавались очищенные от артефактов сырые сигналы ЭЭГ от выбранных каналов. Используя эти сигналы, сеть настраивалась на классификацию воображаемых команд, соответствующих сжатию и разжатию кисти левой и правой рук. На выходе сети формировался индекс воображаемой команды, на распознанное движение. Тестирование показало самые низкие точности классификации. В автономных сессиях, когда классификация паттернов воображаемых команд проводилась по одной попытке, вероятность успешной классификации команд при случайном пороге 50% в среднем составила  $54 \pm 10\%$ . После овладения испытуемым навыком обращения с нейроинтерфейсом средняя точность составляла  $63 \pm 6\%$ , а максимальная достигала 72% (таблица 1).

Тестирование классификаторов на основе римановой геометрии показало результаты несколько ниже, чем у комитетного классификатора. В автономных сессиях, когда классификация ЭЭГ-паттернов воображаемых команд проводилась по одной попытке, вероятность успешной классификации 2-х типов команд (сжатие левой и правой кисти) при случайном пороге 50% в среднем составила  $60 \pm 8\%$ . После овладения испытуемым навыком обращения с нейроинтерфейсом средняя точность составляла  $64 \pm 8\%$ , тогда как максимальная точность классификации достигала 83% (таблица 1).

Нужно отметить, что похожие результаты были получены при сравнительном исследовании подобных типов классификаторов другими исследователями [17].

Результаты тестирования нейроморфного классификатора взяты из работы [16]. Классификатор на СпНС описанной архитектуры показал наибольшую точность при распознавании двух классов ментальных состояний (стресса и расслабления): средняя точность составила  $90\% \pm 8\%$ , а максимальная точность достигала 95% (таблица 1).

Таблица 1. Результаты тестирования классификаторов

	Комитет классификаторов	Сверточная сеть	Риманова геометрия	Нейро-морфная сеть
Средняя точность	70 ± 10%	63 ± 6%	64 ± 8%	90 ± 5%
Максимальная точность	93%	72%	83%	95%

Приведенные результаты показывают относительно низкую точность распознавания у первых трех типов классификаторов и существенно более высокую точность у нейроморфного классификатора. Однако все испытания проводились только для двух классов ПВП. При практическом использовании нейроинтерфейсов с такими классификаторами, например, для управления робототехническими устройствами, требуется распознавать более двух классов ПВП с точностью более 90%. Поэтому для развития этого направления были проведены дополнительные исследования для разработки нейроморфного классификатора, способного распознавать 4 и более ПВП, соответствующие воображаемым командам, с точностью не менее 90%.

**3. Нейроморфный классификатор.** Как видно из таблицы 1, классификатор на основе СпНС показал наибольшую точность классификации ПВП по сравнению с тремя типами других классификаторов. Вероятно, это может быть связано с некоторыми особыми свойствами, наследуемыми от биологических нейронных сетей, которые эффективно решают задачи классификации пространственно-временных сигналов, имеющих вид импульсных потоков. В каждом нейроне биологической сети множественные входные импульсные потоки кодируются в непрерывные величины (мембранные потенциалы) и после суммирования и порогового преобразования декодируются в импульсный поток на выходе нейрона. Импульсные потоки передаются от нейрона к нейрону, прежде всего, по причине экономии энергии.

В классификаторах на СпНС сначала выполняется кодирование входной информации – преобразование непрерывных сигналов в импульсные потоки. При этом часто применяется простой способ кодирования, когда каждому значению сигнала устанавливается определенная частота импульсов. Однако при этом игнорируется временная информация. Более сложные методы используют временное кодирование, например, кодирование с задержкой, фазовое кодирование на уровне одного нейрона или популяционное



кодирование для группы нейронов. Далее работает сама СпНС, на выходах которой формируются импульсные потоки, которые затем декодируются, например, с использованием частотной схемы: выходной нейрон сети с наибольшей частотой генерации импульсов рассматривается как превалирующий класс, соответствующий распознанному состоянию объекта.

Применение СпНС позволяет обойти ограничения ИНС на формальных нейронах за счет улучшения адаптации к новым условиям с сохранением предыдущих полученных навыков. Кроме того, имеет место большая устойчивость разрабатываемых систем к помехам и переменным данным, а также низкое энергопотребления в случае аппаратной реализации СпНС за счет передачи данных импульсными потоками.

Учитывая описанные свойства классификатора на СпНС, был разработан его новый вариант. В отличие от описанного в работе [16] варианта нейроморфного классификатора с простейшими спайковыми нейронами на модели LIF, в предлагаемом варианте СпНС строился как множество связанных биоподобных спайковых нейронов, модель которых описывается более сложными системами дифференциальных уравнений (модель Ижикевича). Предполагалось, что в таком варианте СпНС возможно достижение большей точности классификации, чем в предыдущем варианте с упрощенными нейронами.

Так, модель LIF, хотя требует меньше вычислительных затрат, но игнорирует форму потенциала действия и обращает внимание только на выбор времени (тайминг) спайка. Она описывается уравнением [18]:

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t).$$

Потенциал действия немедленно становится  $u_{rest}$  после поджигания. Абсолютный невосприимчивый период может быть определен как  $u = -u_{abs}$  в течение периода после поджигания и затем потенциал действия устанавливается как  $u_{rest}$ .

По сравнению с этой простейшей моделью биоподобного нейрона, модель Ижикевича обеспечивает баланс между стоимостью вычислений и биологической реальностью. Она дает возможность описать много разных типов поджиганий. Эта феноменологическая модель описывает динамику мембранного потенциала. По критериям биологической релевантности она приближается к известной модели Ходжкина – Хаксли, но требует при этом гораздо меньше вычислительных ресурсов, что особенно важно при моделировании

нейронных сетей больших размеров. Модель описывается следующей системой уравнений [19]:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - I(t),$$

$$\frac{du}{dt} = a(bv - u),$$

с дополнительным условием сброса переменных при достижении пика импульса:

$$\text{if}(v \geq +30)\text{then}(v \leftarrow c) \wedge (u \leftarrow u + d),$$

где  $v$  – трансмембранный потенциал,  $u$  – переменная восстановления,  $a$ ,  $b$ ,  $c$ ,  $d$  – параметры модели,  $I(t)$  – ток, приложенный к мембране извне. При достижении потенциалом 30 мВ регистрируется спайк, и переменные сбрасываются до значений, описанных в уравнениях. В данной работе использовались следующие значения:  $a = 0.02$ ;  $b = 0.2$ ;  $c = -65$  и  $d = 8$ . При таких параметрах в отсутствии внешнего воздействия нейрон находится в состоянии покоя, а при приложении внешнего постоянного тока генерирует регулярные импульсы, что характерно, например, для кортикальных нейронов.

В первом уравнении при моделировании сети нейронов внешний ток может быть представлен следующим образом:

$$I(t) = \varepsilon(t) + I_{syn}(t) + I_{stim}(t),$$

где  $\xi(t)$  – некоррелированный белый гауссовский шум со средним 0 и стандартным отклонением  $D$ ,  $I_{syn}(t)$  – синаптический ток,  $I_{stim}(t)$  – ток внешней стимуляции. В качестве внешней стимуляции может использоваться последовательность импульсов с частотой 10 Гц, с длительностью 3 мс и амплитудой, достаточной для возбуждения нейрона.

Синаптический ток вычислялся как сумма взвешенных выходных сигналов нейронов, посылающих связи к данному нейрону:

$$I_{syn}(t) = \sum_j g_j w_{ij} y_{ij}(t),$$

где  $g_j$  – коэффициент трансформации выходного сигнала нейрона  $j$  в синаптический ток (в работе использовались значения  $g = 20$  для возбуждающих нейронов и  $g = -20$  для тормозных),  $w_{ij}$  – вес связи,

идущей от пресинаптического нейрона  $j$  к постсинаптическому нейрону  $i$ ,  $y_{ij}(t)$  – выходной сигнал пресинаптического нейрона, в качестве которого подразумевается порция нейромедиатора, выбрасываемого в синапсах с каждым импульсом.

Первая задача, возникшая при проектировании нейроинтерфейса с классификатором на СпНС – импульсное кодирование непрерывных сигналов на входах – была решена выбором способа временного кодирования [20, 21].

Перед подачей данных в СпНС происходит преобразование спайков. Уравнение (1) обеспечивает вычисление порога, где  $M(dif)$  и  $Std(dif)$  означают среднее и стандартное отклонения временной разницы сигналов ( $dif$ ). Значение  $dif$  конкретного канала ЭЭГ рассчитывается путем вычитания амплитуды в момент времени  $t$  на амплитуду в момент  $t - 1$ . Пороговый коэффициент  $f$  устанавливается пользователем (в наших экспериментах он был установлен на 0,5). После расчета  $Tr$  в соответствии с (1) сравниваются  $dif$  амплитуды в каждый момент времени. Если амплитуда  $dif$  превышает  $Tr$  в момент времени  $t$ , излучается возбуждающий спайк, а если амплитуда  $dif$  падает ниже  $Tr$ , излучается тормозной спайк.

$$Tr = M(dif) + f \times Std(dif). \quad (1)$$

Как показано на рисунке 1, где пики исходного сигнала представлены серией пиков, этот алгоритм кодирования придает особое значение флуктуациям амплитуды, позволяя сохранять характерные события в сигнале ЭЭГ. Этот метод может отфильтровать мельчайшие колебания, вызванные шумом сигнала.

Решение второй задачи – синтез структуры классификатора на СпНС – привело к выбору трехслойной прямонаправленной СпНС (рисунок 2). Первый слой сети включал 2 нейрона, которые принимали сигналы от двух выбранных предварительно каналов. Второй слой включал 5 и более нейронов, число которых определялось экспериментально и зависело от числа примеров, используемых при обучении классификатора. Третий слой сети в нашем случае включал 4 нейрона, что соответствовало количеству классов распознаваемых состояний. Задача обучения, т.е. настройки параметров СпНС с такой архитектурой, решалась на основе метода Supervised Spike Timing Dependent Plasticity (SSTDP) [22].

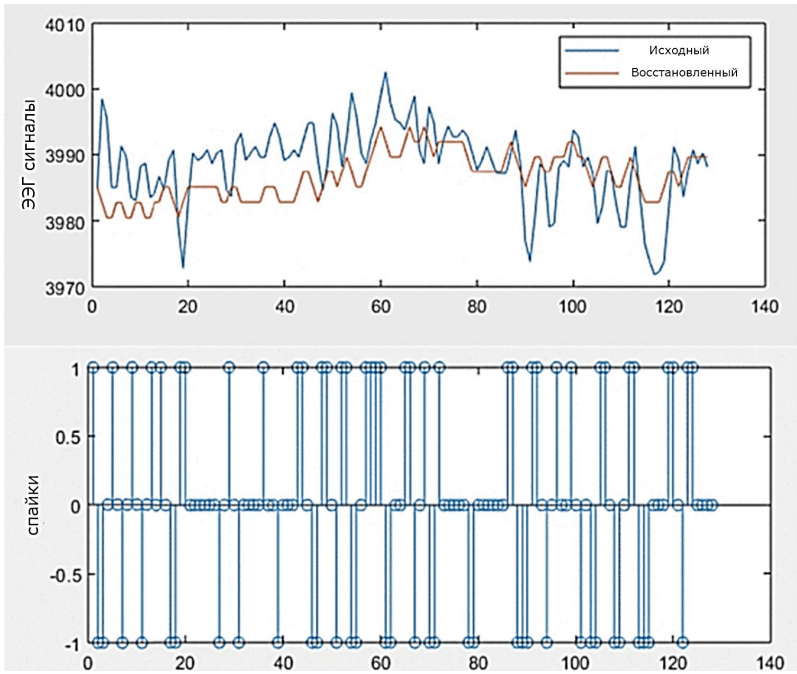


Рис. 1. Спайковое представление канала, экстрагированного из сигнала ЭЭГ. Верхние графики иллюстрируют оригинальный сигнал ЭЭГ (голубой) и реконструированный сигнал (красный)

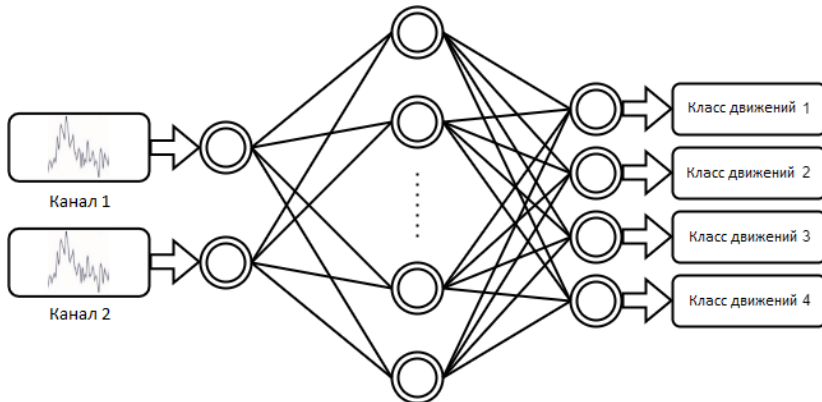


Рис. 2. Структура нейроморфного классификатора

Обучение и тестирование классификатора на СпНС проводилось на ПВП, содержащих волны вызванных потенциалов Р300, возникающих при поочередном отображении на мониторе 4-х стрелок, соответствующих направлениям движений, воображаемых испытуемым. Тестирование показало среднюю точность при 4-х классах ПВП  $90 \pm 5\%$ , а максимальную точность – до 95%, что в определенной степени соответствует нейроморфному классификатору, но настроенному на распознавание только двух классов ПВП (таблица 1).

**4. Пример применения нейроморфного классификатора.** Как показано в предыдущей секции, разработанный нейроморфный классификатор продемонстрировал достаточную точность и способность работать в реальном времени. На следующем этапе он был исследован на возможность использования в системе прямого бесконтактного управления мобильным роботом. Для практической реализации системы управления использовалась обратная связь в визуальной форме, которая позволяла пользователю научиться в процессе тренинга управлять роботом путем генерации воображаемых моторных команд. При этом требовалось, чтобы задержки ответов системы были достаточно малыми, чтобы мозг пользователя смог правильно ассоциировать ответы с событиями. Показано, что максимальный период задержки при формировании сигналов, определяющих распознанное состояние, не должен превышать 150 мс, что позволяет использовать обратную связь при работе оператора. Для выполнения этого условия в системе была реализована многопоточность, позволяющая записывать и обрабатывать одновременно несколько сигналов в заданном интервале времени без потери данных и выполнять онлайн классификацию.

Предварительные исследования, проведенные авторами этой работы, заложили основу для разработки универсальной структуры системы бесконтактного управления роботами с использованием классификации ПВП, содержащих сигналы ЭЭГ, регистрируемые в разных отведениях сенсомоторной зоны коры мозга [23, 24, 25]. На рисунке 3 представлена структура такой системы для управления передвижениями робота в виде мобильной платформы. ЭЭГ-прибор регистрирует текущие сигналы ЭЭГ по выбранным каналам, а нейроинтерфейс на основе классификатора ПВП производит предобработку данных для определения ЭЭГ-паттернов соответствующих воображаемым командам, после чего происходит расчёт характерных признаков этих паттернов и выполняется

их классификация, формируя на выходе класс движения, к которому относится воображаемая команда. Распознанные команды подаются на блок управления (БУ Перемещения), который формирует команды перемещения мобильной платформы (Вперед, Назад, Влево, Вправо).

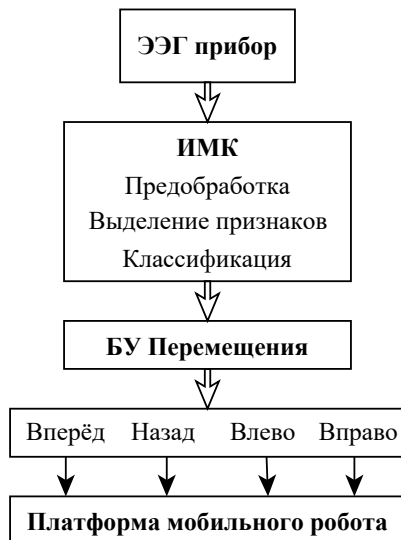


Рис. 3. Структура системы бесконтактного управления мобильным роботом

При проведении экспериментов согласно разработанной методике, пользователям предлагалось последовательно представлять 4 типа направлений движения (вперед, назад, вправо, влево) в соответствии с присвоенными им командами управления перемещением роботизированного устройства. При тренировке пользователи сначала должны мысленно реагировать на выбранные ими стрелки, последовательно отображаемые на мониторе, Тренировка была ориентирована на воображение выбранного направления движения мобильного робота. Тренировочная серия воображаемых направлений движения повторялась несколько раз до тех пор, пока классификатор не начнет уверенно распознавать вызванные потенциалы P300, соответствующие воображаемым направлениям движения. В результате пользователь должен был выполнить несколько десятков воображений направлений движений в одном блоке серии. Количество блоков серии должно соответствовать количеству типов выполняемых движений (движения по 4-м

направлениям). Таким образом, имитировалось прямое управление движением мобильного робота с помощью вызванных потенциалов.

Регистрация ПВП, содержащих сигналы ЭЭГ в разных отведениях выполнялась с помощью нейрогарнитуры «Muse headset 2016». Использовались 4 электрода: TP9, TP10, AF7, AF8 которые были расположены по системе 10-5. Сигналы ЭЭГ длительностью 800 мс регистрировались в полосе частот 1 Гц – 30 Гц. Частота дискретизации составляла 256 Гц. При анализе записей ЭЭГ были исключены артефакты, такие как движения глаз, медленные и быстрые волны, фрагменты сигналов ЭЭГ с амплитудой более 100 мкВ. Однако для последующей обработки использовались только два выбранных канала.

Для определения возможности управления мобильным роботом в соответствии со схемой, структура которой представлена на рисунке 3, был проведен эксперимент, в котором операторы, используя специальную нейрогарнитуру, управляли виртуальной моделью мобильной платформы.

Для управления был разработан программный модуль, преобразующий распознанные с помощью нейроинтерфейса воображаемые направления движения по стрелке на мониторе (рисунок 4) в команды управления виртуальной моделью робота в соответствии с таблицей 2. При управлении движениями: Вперед, Назад, Вправо, Влево было возможно выполнять воображение выбранного движения с заданным шагом (вперед или назад – 10 см, вправо или влево 10 градусов) неоднократно для выполнения дискретного движения или поворота мобильного робота. На этом этапе испытуемые некоторое время учились управлять моделью мобильного робота, многократно воображая нужные направления движения и повороты. Первые попытки управления движением робота показали, что для уверенного прохождения трассы было недостаточно распознавания одной попытки воображения выбранного действия, и робот мог уйти в сторону от трассы. Это было связано с недостаточной точностью распознавания движений в реальном времени по одной попытке. Далее удалось несколько улучшить результат прохождения трассы за счет усреднения трех попыток воображения выбранного движения.

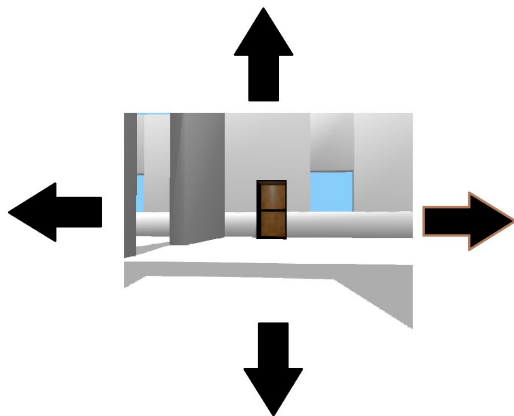


Рис. 4. Стрелки на мониторе для выбора направления движений

Таблица 2. Соответствие отображаемых символов и воображаемых и команд управления

Номер команды	Отображаемые на мониторе символы	Воображаемые команды
1	Стрелка вправо	Вправо
2	Стрелка влево	Влево
3	Стрелка вверх	Вперед
4	Стрелка вниз	Назад

Эксперимент по управлению виртуальной моделью мобильного робота проводился в среде Gazebo [26, 27] с использованием предварительно обученного классификатора на нейроморфной сети. Для эксперимента была создана модель робота на трехколесном шасси (два колеса ведущих, одно – флюгерное). Модель имитировала сервисного робота для развозки лекарств и пищи обездвиженным пациентам. На рисунке 5 показан вид модели, выполненной в программе «gobot» и функционирующей в среде Gazebo. Здесь же показана траектория движения робота при бесконтактном управлении.

В эксперименте участвовали 5 здоровых испытуемых. Предварительно производилось обучение классификатора на распознавание ПВП, соответствующих выбранным пользователем воображаемым направлениям движения по стрелкам на мониторе отдельно для каждого из испытуемых.

При этом была достигнута средняя точность классификации в онлайн режиме – 90%. Предполагается, что при тщательной настройке классификатора, они должны улучшиться.



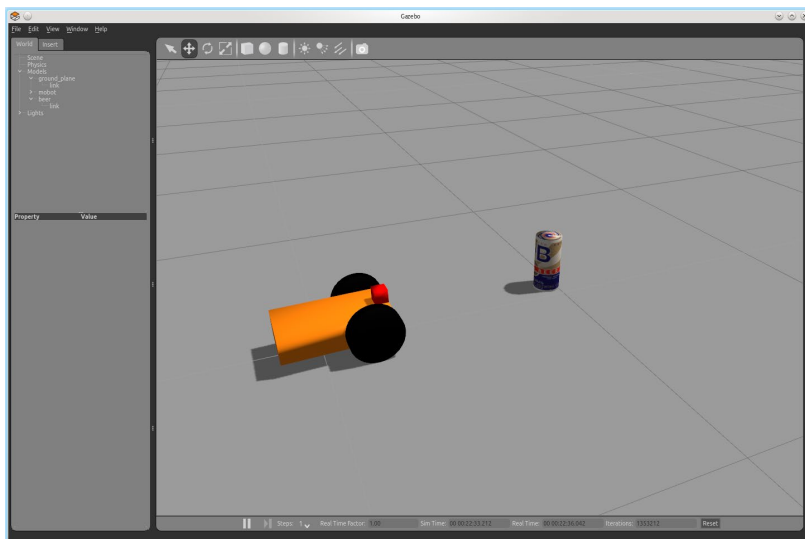


Рис. 5. Робот в среде Gazebo

**5. Заключение.** Классификация пространственно-временных паттернов, содержащих совокупность сигналов, имеющих определенную продолжительность во времени и снятых в разных точках пространства, выявляет ситуации процессов или действия объектов. Это позволяет эффективно решать задачи контроля сложных процессов, а также диагностики и управления устройствами со сложным поведением.

Сравнительное исследование различных методов классификации пространственно-временных паттернов, проведенное на данных биоэлектрической активности мозга, соответствующих двум классам ментальных состояний (например, воображаемых движений), показало, что все тестированные классификаторы после обучения способны различать эти состояния, но имеют разную точность. Классификатор на нейроморфных сетях показал лучшую точность по сравнению с классификаторами на нейронных сетях, методе опорных векторов и римановой геометрии. Однако эта точность показана только на двух классах состояний, что является недостаточным для практического использования, например, при управлении устройствами. Кроме того, этот классификатор формирует ответы с относительно большой задержкой, что затрудняет его использование в контурах обратной связи с оператором. Это привело

к необходимости разработки нового варианта нейроморфного классификатора, более пригодного для управляющих приложений.

Разработанный нейроморфный классификатор пространственно-временных паттернов позволяет распознавать 4 и более классов состояний с точностью  $90 \pm 5\%$ . Чтобы удовлетворять условию реального времени, максимальный период задержки при формировании сигналов, определяющих распознанное состояние, не должен превышать 150 мс, что позволяет использовать обратную связь при работе оператора. Для этого в системе управления реализована многопоточность, позволяющая записывать и обрабатывать одновременно несколько сигналов в заданном интервале времени без потери данных и выполнять онлайн классификацию.

Эксперимент показал, что 4 из 5-ти испытуемых после нескольких тренировочных сессий справились с управлением мобильной платформой в режиме прямого командного управления. Можно сделать предварительный вывод о том, что реальный мобильный робот такого типа может использоваться для подвоза по командам лежачего человека лекарства и воды, а также отвоза использованных приборов. Дополнительное оснащение платформы манипулятором необходимо, чтобы реализовать погрузку и разгрузку требуемых предметов. Решение этих задач позволит создать эффективные роботизированные средства, которые обеспечат людям с ограниченными двигательными функциями возможности управления робототехническими устройствами, используя сигналы мозга, что может значительно улучшить их функционирование в мире здоровых людей.

Нужно отметить, что на практике решение задач на импульсных моделях нейронов затруднено из-за высокой вычислительной сложности при их реализации, основанной на решении дифференциальных уравнений. Для снижения вычислительной сложности и ускорения работы импульсных нейронов можно применять различные приближения решений и даже аппаратные реализации СпНС. Эти подходы предполагается исследовать в дальнейшем.

### Литература

1. Лисовский А.Л. Применение нейросетевых технологий для разработки систем управления. Стратегические решения и риск-менеджмент. 2020. Т. 11. № 4. С. 378–389. DOI: 10.17747/2618-947X-923.
2. Благовещенский В.Г., Благовещенский И.Г., Благовещенская М.М., Адноворцев А.М., Головин В.В. Управление технологическими процессами производства кондитерских изделий с использованием нейросетевого регулятора. Труды Всероссийской НТК «Информатизация и автоматизация

- в пищевой промышленности». Курск: Изд-во ЗАО «Университетские книги», 2022. С. 78–83.
3. Ульев А.Д., Розалиев В.Л., Заболеева-Зотова А.В., Орлова Ю.А. Интеллектуальная система видеонаблюдения за поведением человека // Искусственный интеллект и принятие решений. 2020. № 4. С. 21–32. DOI: 10.14357/20718594200403.
  4. Богущ Р.П., Захарова И.Ю. Алгоритм сопровождения людей на видеопоследовательностях с использованием сверточных нейронных сетей для видеонаблюдения внутри помещений // Компьютерная оптика. 2020. Т. 44. № 1. С. 109–116. DOI: 10.18287/2412-6179-CO-565.
  5. Brunner C., Birbaumer N., Blankertz B., Guger C., Kubler A., Mattia D., del R. Millan J., Miralles F., Nijholt A., Opisso E., Ramsey N., Salomon P., Muller-Putz G.R. BNCI Horizon 2020: towards a roadmap for the BCI community // Brain-Computer Interfaces. 2015. vol. 2. no. 1. pp. 1–10. DOI: 10.1080/2326263X.2015.1008956.
  6. Sharmila A. Hybrid control approaches for hands-free high level human-computer interface-a review // Journal of Medical Engineering & Technology. 2021. Т. 45. № 1. pp. 6–13.
  7. Diez P. Smart Wheelchairs and BCI. Mobile Assistive Technologies // Academic Press, 2018. 492 p.
  8. Кагиров И.А., Карпов А.А., Кипяткова И.С., Ключев К.С., Кудрявцев А.И., Кудрявцев И.А., Рюмин Д.А. Интеллектуальный интерфейс для управления роботизированным медицинским экзоскелетом нижних конечностей Remotion // Авиакосмическая и экологическая медицина, 2019. Т. 53. № 5. С. 92–98.
  9. Li Z., Li B., Luo W., Cao J. Design and Implementation of P300 Brain-Controlled Wheelchair with a Developed Wireless DA Converter. International journal of computers & technology. 2023. vol. 23. pp. 93–104. DOI: 10.24297/ijct.v23i.9485.
  10. Yakovlev L., Kaplan A., Sirov N. Gortz, N. BCI-Controlled Motor Imagery Training Can Improve Performance in e-Sports. HCI International 2020-Posters: 22nd International Conference. 2020. pp. 581–586. DOI: 10.1007/978-3-030-50726-8\_76.
  11. Zhu H.Y., Hieu N.Q., Hoang D.T., Nguyen D.N., Lin C.-T. A Human-Centric Metaverse Enabled by Brain-Computer Interface: A Survey. arXiv preprint arXiv:2309.01848. 2023.
  12. Станкевич Л.А., Сонькин К.М., Нагорнова Ж.В., Хоменко Ю.Г., Шемякина Н.В. Классификация электроэнцефалографических паттернов воображаемых движений пальцами руки для разработки интерфейса мозг-компьютер. Труды СПИИРАН. 2015. Т. 3(40). С. 163–182. DOI: 10.15622/sp.40.11.
  13. Stankevich L.A., Sonkin K.M., Shemyakina N.V., Nagornova Zh.V., Khomenko Ju.G., Perets D.S., Koval A.V. EEG Pattern Decoding of Rhythmic Individual Finger Imaginary Movements of one Hand. Human Physiology. 2016. vol. 42. no. 1. pp. 32–42.
  14. Schirrmester R.T., Springenberg J.T., Fiederer L.D.J., Glasstetter M., Eggenberger K., Tangermann M., Hutter F., Burgard W., Ball T. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. arXiv:1703.05051v5. 2018.
  15. Congedo M., Barachant A., Bhatia R. Riemannian geometry for EEG-based brain-computer interfaces: a primer and a review // Brain-Computer Interfaces. 2017. vol. 4. no. 3. pp. 155–174. DOI: 10.1080/2326263X.2017.1297192.
  16. Weerasinghe M.M., Espinosa-Ramos J.I., Wang G.Y., Parry D. Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling // IEEE Access. 2021. vol. 10. pp. 117338–117348. DOI: 10.1109/ACCESS.2021.3099492.

17. Капралов Н.В., Нагорнова Ж.В., Шемякина Н.В. Методы классификации ЭЭГ-паттернов воображаемых движений // Информатика и автоматизация. 2021. Т. 20. № 1. С. 94–132. DOI: 10.15622/ia.2021.20.1.4.
18. Gerstner W., Kistler W.M., Naud R., Paninski L. Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge: Cambridge University Press, 2014. 578 p.
19. Izhikevich E.M. Simple model of spiking neurons. IEEE Trans. Neural Networks. 2003. vol. 14. no. 6. pp. 1569–1572. DOI: 10.1109/TNN.2003.820440.
20. Cui Y., Ahmad S., Hawkins J. The HTM spatial pooler—a neocortical algorithm for online sparse distributed coding // Frontiers in computational neuroscience. 2017. vol. 11. DOI: 10.3389/fncom.2017.00111.
21. Auge D., Hille J., Mueller E., Knoll A. A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks // Neural Processing Letters. 2021. vol. 53. no. 6. pp. 4693–4710. DOI: 10.1007/s11063-021-10562-2.
22. Liu F., Zhao W., Chen Y., Wang Z., Yang T., Jiang L. SSTDP: Supervised Spike Timing Dependent Plasticity for Efficient Spiking Neural Network Training. Frontiers in Neuroscience. 2021. vol. 15. DOI: 10.3389/fnins.2021.756876.
23. Станкевич Л.А., Гунделах Ф.В. Управление роботом с использованием интерфейса «мозг-компьютер» // Робототехника и техническая кибернетика. 2017. № 2(15). С. 52–56.
24. Гунделах Ф.В., Станкевич Л.А., Сонькин К.М., Шемякина Н.В., Нагорнова Ж.В. Применение интерфейсов «мозг-компьютер» в ассистивных технологиях // Труды СПИИРАН. 2020. Т. 19. № 2. С. 277–301.
25. Gundelakh F., Stankevich L., Kapralov N.V., Ekimovskii J.V. Cyber-Physical System Control Based on Brain-Computer Interfaces. Springer International Publishing, 2020. pp. 458–469.
26. Tutorial: ROS integration overview. URL: [https://classic.gazebosim.org/tutorials?tut=ros\\_overview](https://classic.gazebosim.org/tutorials?tut=ros_overview) (дата обращения: 09.12.2023).
27. Ackerman E. Latest Version of Gazebo Simulator Makes It Easier Than Ever to Not Build a Robot. IEEE Spectrum. IEEE. 2016.

**Гунделах Филипп Викторович** — аспирант, кафедра мехатроники и роботостроения, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ). Область научных интересов: робототехника, интеллектуальные системы управления, обработка данных, интерфейсы «мозг-компьютер». Число научных публикаций — 26. [f.gundelakh@yandex.su](mailto:f.gundelakh@yandex.su); улица Политехническая, 21, 195251, Санкт-Петербург, Россия; р.т.: +7(812)297-6780.

**Станкевич Лев Александрович** — канд. техн. наук, доцент, высшая школа автоматизации и робототехники, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ). Область научных интересов: искусственный интеллект, когнитивные системы, антропоморфная робототехника, интерфейсы «мозг-компьютер». Число научных публикаций — 280. [Stankevich\\_lev@inbox.ru](mailto:Stankevich_lev@inbox.ru); улица Политехническая, 21, 195251, Санкт-Петербург, Россия; р.т.: +7(812)552-9686.

**Поддержка исследований.** Исследование выполнено за счет гранта Российского научного фонда № 23-21-00287, <https://rscf.ru/project/23-21-00287>.

F. GUNDELAKH, L. STANKEVICH  
**CLASSIFICATION OF SPATIAL TEMPORAL PATTERNS BASED  
ON NEUROMORPHIC NETWORKS**

*Gundelakh F., Stankevich L. Classification of Spatial Temporal Patterns Based on Neuromorphic Networks.*

**Abstract.** This work is devoted to the problems of developing neuromorphic classifiers of spatiotemporal patterns, as well as their application in neurointerfaces. Classifiers of spatiotemporal patterns based on neural networks, support vector machines, deep neural networks, and Riemannian geometry are considered. A comparative study of these classifiers is carried out in the plane of the accuracy of multiclass recognition of electroencephalographic signals showing time-dependent bioelectrical activity in different areas of the brain during the imagination of different movements. It is shown that such classifiers can provide an accuracy of 60-80% when recognizing from two to four classes of imaginary movements. A new type of classifier based on a neuromorphic network, based on the biosimilar neurons built on the Izhikevich model, is proposed. The network processes input spike sequences and generates pulse streams of different frequencies at the outputs. The network is trained using the Supervised STDP algorithm based on labeled information containing examples of the correct recognition of the required pattern classes. The recognized pattern class is determined by the maximum frequency of the output sequence. The neuromorphic classifier showed an average classification accuracy of 90% for 4 classes of imaginary commands and a maximum of 95%. By modeling the robot control task in the virtual environment it is shown that such accuracy is sufficient for the effective use of the classifier as part of a non-invasive brain-computer interface for non-contact control of robotic devices.

**Keywords:** spatiotemporal patterns, classification, neural networks, support vectors, Riemannian geometry, neuromorphic networks, neurointerface, electroencephalography, imaginary movements, non-contact control.

## References

1. Lisovsky A.L. Application of neural network technologies for management development of systems. Strategic decisions and risk management. 2020. vol. 11. no. 4. pp. 378–389. DOI: 10.17747/2618-947X-923. (In Russ.).
2. Blagoveshhenskij V.G., Blagoveshhenskij I.G., Blagoveshhenskaja M.M., Adnodvorcev A.M., Golovin V.V. [Control of technological product processes for confection wares using neural network regulator] Upravlenie tehnologicheskimi processami proizvodstva konditerskih izdelij s ispol'zovaniem nejrosetevogo reguljatora [Proceedings of the All-Russian Scientific and Technical Committee «Informatization and automation in the food industry»]. Kursk: Published: Books of Universities, 2022. pp. 78–83.
3. Uliev A.D., Rozaliev V.L., Zaboleeva-Zotova A.V., Orlova Y.A. [An Intelligent Video Surveillance System for Human Behavior]. Iskusstvennyj intellekt i prinjatje reshenij – Artificial intelligence and decision making. 2020. no. 4. pp. 21–32. DOI: 10.14357/20718594200403. (In Russ.).
4. Bohush R.P., Zakharava I.Y. Person tracking algorithm based on convolutional neural network for indoor video surveillance. Computer Optics. 2020. vol. 44. no. 1. pp. 109–116. DOI: 10.18287/2412-6179-CO-565. (In Russ.).
5. Brunner C., Birbaumer N., Blankertz B., Guger C., Kubler A., Mattia D., del R. Millan J., Miralles F., Nijholt A., Opisso E., Ramsey N., Salomon P., Muller-

- Putz G.R. BNCI Horizon 2020: towards a roadmap for the BCI community. *Brain-Computer Interfaces*. 2015. vol. 2. no. 1. pp. 1–10. DOI: 10.1080/2326263X.2015.1008956.
6. Sharmila A. Hybrid control approaches for hands-free high level human–computer interface—a review. *Journal of Medical Engineering & Technology*. 2021. vol. 45. no. 1. pp. 6–13.
  7. Diez P. *Smart Wheelchairs and BCI. Mobile Assistive Technologies*. Academic Press, 2018. 492 p.
  8. Kagirov I.A., Karpov A.A., Kipyatkova I.S., Klyuzhev K.S., Kudryavtsev A.I., Kudryavtsev I.A., Ryumin D.A. [Intellectual Interface to Control a Robotic Medical Exoskeleton of the Lower Limbs «Remotion»]. *Aviakosmicheskaja i jekologicheskaja medicina – Aviacosmos and ecological medicine*. 2019. vol. 53. no. 5. pp. 92–98. (In Russ.).
  9. Li Z., Li B., Luo W., Cao J. Design and Implementation of P300 Brain-Controlled Wheelchair with a Developed Wireless DA Converter. *International journal of computers & technology*. 2023. vol. 23. pp. 93–104. DOI: 10.24297/ijct.v23i.9485.
  10. Yakovlev L., Kaplan A., Sirov N., Gortz, N. BCI-Controlled Motor Imagery Training Can Improve Performance in e-Sports. *HCI International 2020-Posters: 22nd International Conference*. 2020. pp. 581–586. DOI: 10.1007/978-3-030-50726-8\_76.
  11. Zhu H.Y., Hieu N.Q., Hoang D.T., Nguyen D.N., Lin C.-T. A Human-Centric Metaverse Enabled by Brain-Computer Interface: A Survey. *arXiv preprint arXiv:2309.01848*. 2023.
  12. Stankevich L., Sonkin K., Nagornova Z., Khomenko J., Shemyakina N. Classification of Electroencephalographic Patterns of Imaginary One-hand Finger Movements for Brain-Computer Interface Development. *SPIIRAS Proceedings*. 2015. vol. 3(40). pp. 163–182. DOI: 10.15622/sp.40.11. (In Russ.).
  13. Stankevich L.A., Sonkin K.M., Shemyakina N.V., Nagornova Zh.V., Khomenko Ju.G., Perets D.S., Koval A.V. EEG Pattern Decoding of Rhythmic Individual Finger Imaginary Movements of one Hand. *Human Physiology*. 2016. vol. 42. no. 1. pp. 32–42.
  14. Schirrmester R.T., Springenberg J.T., Fiederer L.D.J., Glasstetter M., Eggenesperger K., Tangermann M., Hutter F., Burgard W., Ball T. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. *arXiv:1703.05051v5*. 2018.
  15. Congedo M., Barachant A., Bhatia R. Riemannian geometry for EEG-based brain-computer interfaces: a primer and a review // *Brain-Computer Interfaces*. 2017. vol. 4. no. 3. pp. 155–174. DOI: 10.1080/2326263X.2017.1297192.
  16. Weerasinghe M.M., Espinosa-Ramos J.I., Wang G.Y., Parry D. Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling. *IEEE Access*. 2021. vol. 10. pp. 117338–117348. DOI: 10.1109/ACCESS.2021.3099492.
  17. Kapralov N., Nagornova Z., Shemyakina N. Classification Methods for EEG Patterns of Imaginary Movements. *Informatics and Automation*. 2021. vol. 20. no. 1. pp. 94–132. DOI: 10.15622/ia.2021.20.1.4. (In Russ.).
  18. Gerstner W., Kistler W.M., Naud R., Paninski L. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge: Cambridge University Press, 2014. 578 p.
  19. Izhikevich E.M. Simple model of spiking neurons. *IEEE Trans. Neural Networks*. 2003. vol. 14. no. 6. pp. 1569–1572. DOI: 10.1109/TNN.2003.820440.
  20. Cui Y., Ahmad S., Hawkins J. The HTM spatial pooler—a neocortical algorithm for online sparse distributed coding. *Frontiers in computational neuroscience*. 2017. vol. 11. DOI: 10.3389/fncom.2017.00111.

21. Auge D., Hille J., Mueller E., Knoll A. A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks. *Neural Processing Letters*. 2021. vol. 53. no. 6. pp. 4693–4710. DOI: 10.1007/s11063-021-10562-2.
22. Liu F., Zhao W., Chen Y., Wang Z., Yang T., Jiang L. SSTDP: Supervised Spike Timing Dependent Plasticity for Efficient Spiking Neural Network Training. *Frontiers in Neuroscience*. 2021. vol. 15. DOI: 10.3389/fnins.2021.756876.
23. Stankevich L.A. Gundelakh F.V. Robot control with use brain-computer interface. *Robotics and technical cybernetics*. 2017. no. 2(15). pp. 52–56. (In Russ.).
24. Gundelakh F., Stankevich L., Sonkin K., Nagornova G., Shemyakina N. Application of Brain-computer Interfaces in Assistive Technologies. *SPIIRAS Proceedings*. 2020. vol. 19. no. 2. pp. 277–301. (In Russ.).
25. Gundelakh F., Stankevich L., Kapralov N.V., Ekimovski J.V. *Cyber-Physical System Control Based on Brain-Computer Interfaces*. Springer International Publishing, 2020. pp. 458–469.
26. Tutorial: ROS integration overview. Available at: [https://classic.gazebosim.org/tutorials?tut=ros\\_overview](https://classic.gazebosim.org/tutorials?tut=ros_overview) (accessed: 09.12.2023).
27. Ackerman E. Latest Version of Gazebo Simulator Makes It Easier Than Ever to Not Build a Robot. *IEEE Spectrum*. IEEE. 2016.

**Gundelakh Filipp** — Ph.d. student, Mechatronics and robotics department, Peter the Great Saint-Petersburg Polytechnic University (SPbPU). Research interests: robotics, intellectual control systems, data processing, brain-computer interfaces. The number of publications — 26. [f.gundelakh@yandex.su](mailto:f.gundelakh@yandex.su); 21, Polytechnic St., 195251, St. Petersburg, Russia; office phone: +7(812)297-6780.

**Stankevich Lev** — Ph.D., Associate professor, High school of automation and robotics, Peter the Great Saint-Petersburg Polytechnic University (SPbPU). Research interests: artificial intelligence, cognitive systems, anthropomorphic robotics, brain-computer interfaces. The number of publications — 280. [Stankevich\\_lev@inbox.ru](mailto:Stankevich_lev@inbox.ru); 21, Polytechnic St., 195251, St. Petersburg, Russia; office phone: +7(812)552-9686.

**Acknowledgements.** This research is supported by a grant of the Russian Science Foundation (RCF) No. 23-21-00287, <https://rscf.ru/en/project/23-21-00287>.

А.Д. ОБУХОВ, Д.Л. ДЕДОВ, Д.В. ТЕСЕЛКИН, А.А. ВОЛКОВ,  
А.О. НАЗАРОВА

## БЕССТРЕССОВЫЙ АЛГОРИТМ УПРАВЛЕНИЯ БЕГОВЫМИ ПЛАТФОРМАМИ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

*Обухов А.Д., Дедов Д.Л., Теселкин Д.В., Волков А.А., Назарова А.О. Бесстрессовый алгоритм управления беговыми платформами на основе нейросетевых технологий.*

**Аннотация.** В статье рассматривается задача прогнозирования скорости человека с использованием нейросетевых технологий и компьютерного зрения для минимизации запаздывания в системах управления беговыми платформами, приводящего к риску для здоровья пользователя. Для ее решения разработан бесстрессовый алгоритм, включающий прогнозирование положения и скорости пользователя на беговой платформе, включающий процедуру расчета скорости беговой платформы на основе анализа положения и характера движения пользователя, схему сбора и обработки данных для обучения нейросетевых методов, процедуру определения необходимого количества прогнозируемых кадров для устранения запаздывания. Научная новизна исследования состоит в разработке алгоритма управления беговыми платформами, объединяющего технологии компьютерного зрения для распознавания модели тела пользователя платформы, нейронные сети и методы машинного обучения для определения итоговой скорости человека на основе объединения данных о положении человека в кадре, текущей и прогнозируемой скорости человека. Предложенный алгоритм реализован с использованием библиотек Python, проведена его апробация в ходе экспериментальных исследований при анализе предшествующих 10 и 15 кадров для прогнозирования 10 и 15 следующих кадров. В результате сравнения алгоритмов машинного обучения (линейная регрессия, дерево решений, случайный лес, многослойные, сверточные и рекуррентные нейронные сети) при различных величинах длин анализируемых и прогнозируемых кадров наилучшую точность при прогнозировании положения показал алгоритм RandomForestRegressor, а при определении текущей скорости – плотные многослойные нейронные сети. Проведены экспериментальные исследования по применению разработанного алгоритма и моделей для определения скорости человека (при прогнозе в диапазоне 10-15 кадров получена точность более 90%), а также по их интеграции в систему управления беговой платформой. Испытания показали работоспособность предложенного подхода и корректность работы системы в реальных условиях. Разработанный алгоритм позволяет не использовать чувствительные к помехам датчики, требующие закрепления на теле человека, а прогнозировать действия пользователя за счет анализа всех точек тела человека для снижения запаздывания в различных человеко-машинных системах.

**Ключевые слова:** беговые платформы, нейросетевые технологии, бесстрессовый алгоритм управления, машинное обучение.

**1. Введение.** Одним из распространенных устройств имитации физических нагрузок являются активные (управляемые) беговые платформы, основанные на беговых дорожках. Они успешно применяются для опорно-двигательной реабилитации,



профессиональной подготовки, формирования определенного уровня физической нагрузки [1 – 3].

Специфика и качество функционирования активных беговых платформ напрямую зависит от используемых алгоритмов управления. При их реализации определяющую роль играет пользователь, постоянно создающий труднопрогнозируемые воздействия на систему управления, которая должна своевременно адаптироваться под действия человека и менять параметры или режимы своей работы.

Процесс формирования управляющей команды для беговой платформы основан на получении и анализе данных о движении пользователя, который осуществляется в реальном времени в процессе работы платформы. Основной проблемой в этой предметной области является запаздывание [1], так как время между изменением положения человека, реакцией системы управления и дальнейшим воздействием на положение беговой платформы может оказаться значительным. Возникающая задержка между действием пользователя и ответной реакцией беговой платформой затрудняет плавное управление и приводит к различным негативным эффектам (рывкам, резкому ускорению или торможению), что приводит к значительному стрессовому воздействию на пользователя. При комбинации беговой платформы с системами виртуальной реальности общий негативный эффект значительно усиливается [4].

Взаимодействие с платформой вызывает у пользователя определенный стресс, в котором можно выделить положительную (эустресс) и отрицательную (дистресс) формы [5]. Эустресс выражается в необходимости для человека адаптироваться к возникающим физическим нагрузкам, что приводит к формированию необходимого уровня физической подготовки. С другой стороны, наличие запаздывания и несоответствие между скоростью движения дорожки и скоростью перемещения человека приводит к появлению дополнительных отрицательных форм стресса, отвлекающих пользователя от процесса взаимодействия с виртуальной реальностью и усложняющих выполнение поставленных перед ним задач. Таким образом, под бесстрессовым алгоритмом управления беговыми платформами будем понимать алгоритм, при использовании которого система управления будет минимизировать воздействие дистресса для пользователя платформы, не оказывая влияния на эустресс. Тогда целью исследования будет разработка такого алгоритма.

Анализ исследований, в которых рассматривалось устранение дискомфорта пользователя при использовании активных беговых платформ, показал, что в настоящее время многие алгоритмы

управления такими устройствами демонстрируют существенный уровень запаздывания [6 – 9]. Величина запаздывания варьируется от 0,43 до 2 секунд [1]. Так, в работе [6] с минимальным запаздыванием 0,43-0,57 при скорости в 1,2 м/с смещение пользователя может достигать 0,64 м, что для общепринятой длины таких дорожек (от 1 до 2 метров) является значительной величиной. Если оценивать средние показатели (например, системы из работ [7 – 9]), то запаздывание составляет от 1,5 до 2 секунд, а величина смещения составляет от 1 до 5 метров, что неизбежно приведет к падению с дорожки, риску для здоровья либо экстренному срабатыванию алгоритмов остановки или страховочной системы [10].

Сбор данных о действиях пользователя для принятия своевременных решений может потребовать использования дополнительного оборудования, ограничивающего свободу движений [11]. Поэтому актуальной проблемой в научном и практическом плане является разработка алгоритмов, позволяющих прогнозировать действия тренируемого и обеспечивать компенсацию запаздывания реакции беговой платформы.

Для сбора информации о движении пользователя используются различные устройства (механические и ультразвуковые датчики, устройства трехмерного позиционирования, видеокамеры). В результате анализа их особенностей [12 – 15] сформулированы следующие требования:

- отсутствие на пользователе оборудования, ограничивающего свободу движения;
- невосприимчивость к фоновому электромагнитному излучению тренажера;
- область действия  $\geq 1,5$  м (типовая длина беговой платформы).

Анализ существующих систем отслеживания движений [16, 17] показал, что этим требованиям соответствуют только видеокамеры, если силуэт пользователя явно отличается от фона либо распознан в кадре. Работа с видеопотоком может быть организована с использованием современных моделей машинного обучения, например, MediaPipe, MoveNet, PoseNet и более новых и совершенных алгоритмов [18, 19].

С другой стороны, при использовании систем отслеживания действий человека на основе компьютерного зрения (или иных средств захвата движений) неизбежно возникает эффект запаздывания. Существует несколько факторов, снижающих быстродействие системы [20]:

- время на получение данных от источника (например, камер); время на передачу их в обрабатывающий терминал;

- обработка данных (распознавание объектов с использованием интеллектуальных алгоритмов и моделей машинного обучения);
- передача результата в систему управления; формирование управляющей команды;
- время реакции оборудования или программных инструментов на команду.

В результате запаздывание может длиться до нескольких секунд. Такой показатель быстродействия может привести к несвоевременным реакциям на действия пользователя. Например, когда человек уже находится в другой позиции и выполняет следующую операцию, либо не имеет возможности своевременно среагировать на внештатную ситуацию [8], система может послать сигнал о тех действиях, которые уже не релевантны. Основным способом уменьшения времени запаздывания является прогнозирование движений пользователя, которое может быть реализовано путем применения различных методов регрессии (начиная от обычной линейной и заканчивая более продвинутыми методами машинного обучения).

Предыдущие исследования [20 – 22] показали высокую перспективность и эффективность реализации систем управления на основе анализа текущего положения человека, в том числе, с использованием технологий компьютерного зрения. С другой стороны, разработанные ранее алгоритмы распознавания модели тела человека (например, MediaPipe Objectron для сегментирования ног) не решали проблему с запаздыванием, не прогнозировали состояние пользователя, а также имели трудности с обнаружением человека при высокой скорости движений. При разработке нового алгоритма управления предлагается анализировать и прогнозировать не только положение человека на беговой платформе, но и оценивать его текущую скорость.

Использование методов машинного обучения [23], как основного инструмента поставленных задач, обосновано отсутствием необходимости формирования аналитической зависимости между исходными данными (нормированными координатами точек тела человека) и его текущей скоростью в метрических единицах, что не является тривиальной задачей, так как на эту зависимость влияет масштаб и положение модели тела в кадре, направление и угол движения человека по отношению к камере. Второй причиной использования этой группы методов является возможность сбора большого количества данных о перемещении человека с различной скоростью за счет применения двух подходов [16]:

– запись перемещения человека на камеру с последующим захватом ключевых точек тела; скорость человека при этом определяется путем задания фиксированного значения скорости на беговой дорожке, а также использование ручного старта/стопа для регистрации изменений ходьбы;

– имитационное моделирование перемещения модели человека в виртуальной среде, сохранение анимации в виде видеоряда с последующим захватом положения всех необходимых точек и текущей скорости перемещения, что позволяет за счет изменения скорости анимации варьировать скорость перемещения; для разной скорости перемещения модели могут быть заданы различные анимации (медленная или обычная ходьба, бег), кроме того анимация может быть различна для одной скорости для имитации различной походки.

Два подхода комбинируются после нормирования исходных данных о положении тела человека и объединения информации от двух источников. Количество точек определяется алгоритмами распознавания тела в кадре. Предыдущие исследования показали, что для полной реконструкции модели тела человека достаточно 18 точек [16]. Для захвата точек тела человека возможно использование различных моделей машинного обучения (MediaPipe, MoveNet), а также таких современных решений как TokenPose и их модификаций [24].

Таким образом, основной целью исследования является разработка бесстрессового алгоритма управления беговой платформой, основанного на решении задач определения текущей скорости человека за счет анализа динамики его движений в течение нескольких кадров, а также прогнозирование его скорости и положения, что позволит снизить эффект запаздывания и обеспечить своевременную реакцию системы. В качестве основного инструмента для решения задач определения скорости и ее прогнозирования предлагается использование методов машинного обучения.

**2. Формализация бесстрессового алгоритма управления беговыми платформами на основе нейросетевых технологий.** На основе проведенного анализа можно сделать вывод о необходимости разработки бесстрессовых алгоритмов управления активными беговыми платформами с применением технологий компьютерного зрения и моделей машинного обучения для минимизации запаздывания.

Концепция предлагаемого бесстрессового алгоритма заключается в следующем: необходимо сформировать и проанализировать массивы движений пользователя (полученные

после обработки моделями машинного обучения), после чего выбрать соответствующее действие из библиотеки управляющих команд, наиболее подходящее в рамках текущего состояния человека. Для устранения запаздывания анализа и принятия решений будет производиться не только по набору предыдущих кадров, но с учетом прогноза дальнейших действий человека, путем классификации и прогнозирования его движений и скорости с использованием нейросетевых технологий.

Формализуем алгоритм в виде блок-схемы (рисунок 1) и рассмотрим основные задачи, требующие решения для его практической реализации.

Алгоритм включает обязательные подготовительные операции по работе с беговой платформой [22], включая инициализацию всех необходимых программных модулей, библиотек и непосредственно подключение к драйверу управления беговой платформой, который случит посредником между программным обеспечением системы управления и установленным внутри платформы микроконтроллером. Основной цикл работы алгоритма заключается в запуске платформы, подготовке в отдельном потоке выполнения библиотек по работе с камерой (OpenCV), нейронными сетями (Keras) или алгоритмами машинного обучения (scikit-learn). Следующий шаг алгоритма реализует получение видеопотока и распознавание на нем модели тела человека с набором ключевых точек. Задавая в кадре рабочую зону, соответствующую положению беговой платформы в кадре, можно рассчитать скорость платформы  $s_0$  по относительному положению человека на ней, применив простейшую линейную функцию (аналогично исследованию [20]).

Используя подготовленные модели нейронных сетей или алгоритмы машинного обучения осуществляется решение двух задач: определение скорости человека по последовательности наборов точек и прогнозирование его скорости (путем обработки прогноза положения тела человека на некотором временном интервале). Так как движение человека имеет высокую степень неопределенности и хаотичности, а движение ног в процессе ходьбы то ускоряется, то замедляется, в рамках алгоритма предлагается комбинированный расчет скорости платформы с учетом корректирующих коэффициентов ( $k_1$ ,  $k_2$ ,  $k_3$ ), умноженных, соответственно, на  $s_0$ , текущую скорость человека  $s$  и прогноз скорости  $s_p$ , что позволяет учесть следующие ситуации: пользователь готовится к торможению, резко останавливается, ускоряется.

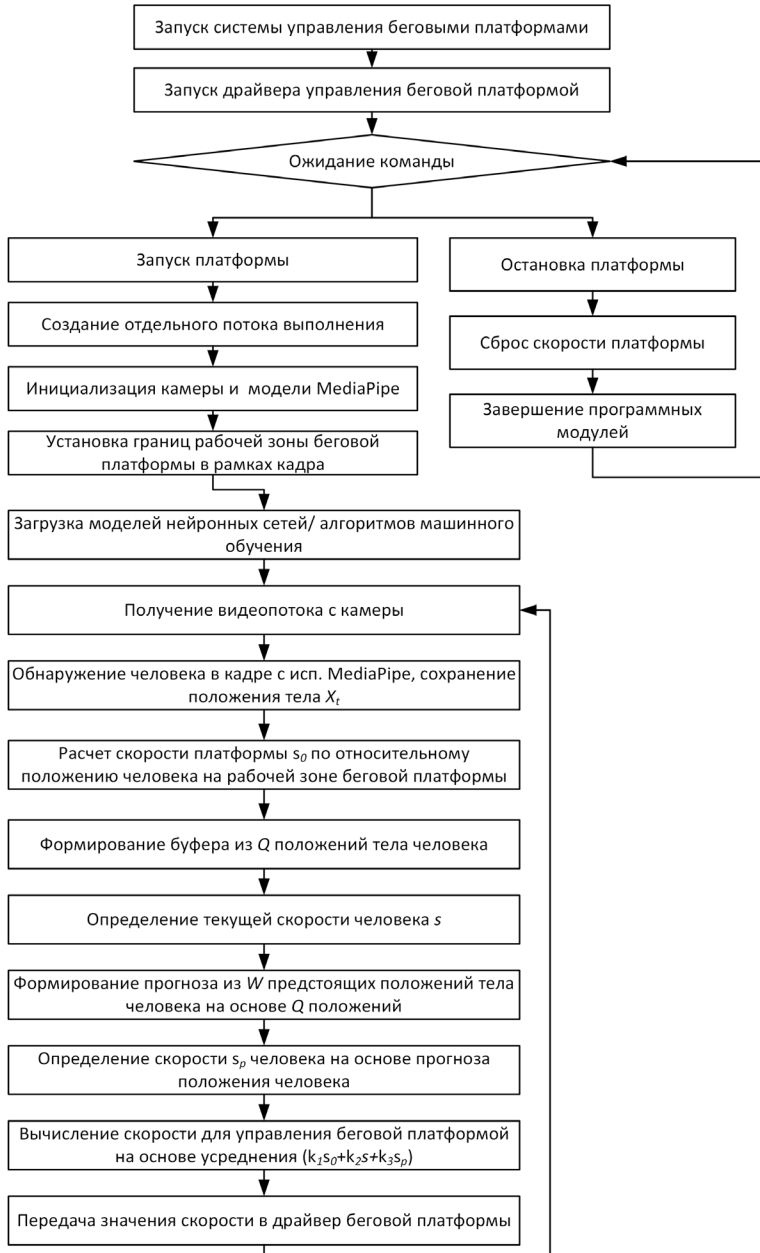


Рис. 1. Бесстрессовый алгоритм управления беговыми платформами

Варьируя корректирующие коэффициенты, возможно настроить систему управления таким образом, чтобы компенсировать недостатки линейной функции управления, а также возможные помехи и неточность работы нейронных сетей для определения скорости. Итоговое значение скорости учитывает текущую динамику движения человека и прогноз на следующий интервал в  $W$  кадров, таким образом снижая запаздывание системы управления и диапазон ее колебаний, тем самым также понижая стресс пользователя, одной из причин которого, как упоминалось выше, являются резкие ускорения и торможения платформы, приводящие к значительному изменению положения человека. Алгоритм также предусматривает остановку платформы в следующих случаях: долгое отсутствие новых кадров из видеопотока, невозможность распознавания человека в кадре, ручная остановка по завершению работы.

Таким образом, для реализации предложенного алгоритма необходимо решить две задачи: определение текущей скорости человека в зависимости от динамики перемещения; прогнозирование его скорости на основе анализа предстоящих положений тела человека.

В формализованном виде получим следующую задачу: необходимо осуществить выбор алгоритма машинного обучения  $A$  и его параметров  $P$ , который аппроксимирует регрессионную зависимость между набором положений тела человека  $\{U_{t-Q+1}, \dots, U_{t-1}, U_t\}$  из  $Q$  количества замеров и его текущей скоростью  $s_i^*$  с минимальной ошибкой  $E$  относительно реальной скорости  $s_i$ .

$$A: \{U_{t-Q+1}, \dots, U_{t-1}, U_t\} \rightarrow s_i, \quad (1)$$

$$E(A) \rightarrow \min.$$

В качестве ошибки  $E$  используется общепринятая для регрессионных задач среднеквадратичная ошибка. Индекс  $t$  обозначает момент времени из множества дискретных интервалов от 0 до окончания работы платформы  $T$ . Каждый элемент  $U_t = \left\{ \langle x_{t,i}, y_{t,i} \rangle \mid i = 0..32 \right\}$  содержит положения множества точек модели тела человека, нормированных относительно центра тела человека, где  $x_{t,i}$  и  $y_{t,i}$  – значения координат по оси X и Y соответственно для  $i$  точки тела человека в момент времени  $t$ .

Для успешного решения этой задачи необходимо определить оптимальный алгоритм машинного обучения, аппроксимирующий зависимость между положениями точек тела человека и его текущей скоростью, что требует анализа и сравнения различных алгоритмов машинного обучения и нейронных сетей. Проведенный анализ выявил, что в этом направлении оправдано использование следующих вариантов алгоритмов [25 – 27]:

- регрессоры на основе деревьев решений (DecisionTreeRegressor): простой в реализации и легко интерпретируемый алгоритм машинного обучения, позволяющий аппроксимировать регрессионную зависимость;

- регрессоры на основе случайного леса (RandomForestRegressor): ансамблевый метод классификации, объединяющий несколько оценщиков (деревьев решений с заданной глубиной ветвления) для повышения итоговой точности;

- многослойные плотные нейронные сети: простые в реализации и универсальные аппроксиматоры;

- рекуррентные нейронные сети типа LSTM: распространенные и эффективные модели при анализе временных рядов;

- многослойные сверточные нейронные сети: позволяют обобщать признаки временных последовательностей и рядов.

Помимо решения задачи (1) в рамках исследования необходимо осуществить прогнозирование скорости человека в некоторый момент времени. Формализуем этапы этой задачи, адаптировав результаты предыдущих исследований.

Пусть задан набор данных  $H_t = \{U_{t-Q+1}, \dots, U_{t-1}, U_t\}$ , соответствующий моменту времени  $t \in [0; T]$ , который обозначим как буфер положений человека с заданной длиной  $Q$ . Тогда в соответствие этому буферу  $H_t$  длиной  $Q$  можно поставить прогноз  $F_t$  длиной  $W$ :  $F_t = \{U_{t+1}, \dots, U_{t-1+W}, U_{t+W}\}$ . Длины  $Q$  и  $W$  могут быть равными или варьироваться, учитывая, что большая длина  $Q$  упрощает прогнозирование благодаря увеличению объема данных, а большая длина  $W$  – усложняет и снижает точность прогноза. С другой стороны, слишком большая величина  $Q$  снижает влияние последних (с близкими к  $t$  индексами) элементов в общем объеме  $H_t$  имеющих большую актуальность и, следовательно, важность.

Сформулируем процесс обучения нейронной сети для прогнозирования следующим образом: заданы входные



$H = \{H_Q, H_{Q+1}, \dots, H_T\}$  и выходные  $F = \{F_Q, F_{Q+1}, \dots, F_{T+W}\}$  данные тренировочной выборки. Тогда для каждого  $t$ -го выходного вектора тренировочной выборки  $F_t$  задан входной вектор  $H_t$ , соответствующий  $Q$  предшествующим набору предыдущих состояний объекта. Таким образом, на вход алгоритма машинного обучения  $A_F$  подается  $Q$  предыдущих состояний объекта, на выходе формируется  $W$  следующих состояний. Тогда задача оптимизации алгоритма  $A_F$  будет состоять в минимизации ошибки отклонения  $E$  прогнозируемых значений от их реальных значений:

$$\begin{aligned} A_F : H &\rightarrow F, \\ E(A_F) &\rightarrow \min. \end{aligned} \tag{2}$$

При решении задачи (2) также используются различные алгоритмы машинного обучения, рассмотренные выше, так как их архитектура предполагает решение обеих задач с изменением только выходного слоя.

На рисунке 2 представлена схема эксперимента, отражающая ход решения задач (1) и (2).

На первом этапе исходный кадр с камеры преобразуется в набор координат ключевых точек тела человека. Далее необходимо осуществить сбор данных о процессе перемещения человека, и сформировать  $Q$  последовательных наборов позиций точек тела человека, поступающих на вход модели нейронной сети прогнозирования. Это позволит снизить запаздывание системы управления на  $W/FPS$  секунд, где  $FPS$  – частота видеоряда, получаемого с камеры.

Для определения размера  $W$  предлагается использовать следующий подход.

На первом этапе оценивается величина запаздывания  $\Omega$  и частота  $FPS$  видеоряда с камеры. Пусть нейросетевая обработка (прогнозирование) за последние  $Q$  кадров (с учетом текущего) занимает время  $\psi$ , тогда общее время работы с 1 кадром обозначим как  $\Psi$ ,  $\Psi = \psi + 1/FPS$ . Время  $\Psi$  всегда необходимо учитывать, так как текущий кадр требуется для осуществления прогнозирования. Тогда для компенсации запаздывания в размере  $\Omega$  требуется осуществить прогнозирование на время  $\Omega + \Psi$ .

Количество кадров, которые алгоритму необходимо прогнозировать, определяется отношением:  $W = (\Omega + \Psi) \cdot FPS$ .

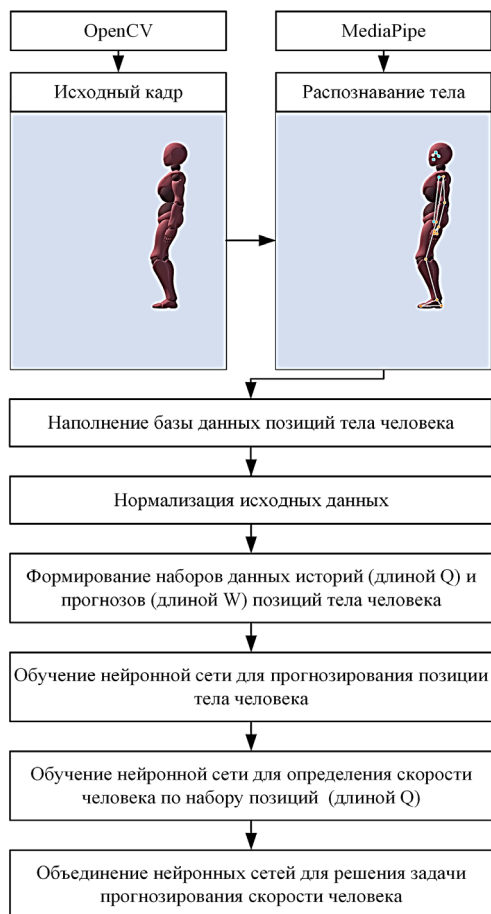


Рис. 2. Схема эксперимента по прогнозированию скорости и положения человека

Предварительные испытания разработанного алгоритма показали необходимость реализации в его рамках дополнительных преобразований исходных данных, а именно их нормализации относительно определенных точек тела человека. Это обусловлено тем, что при сборе информации человек может двигаться в различных частях изображения, поэтому использование изначальных

нормированных относительно размеров кадра координат недостаточно.

Вводится дополнительное преобразование всех точек модели тела человека относительно средней позиции линии бедер (в модели MediaPipe эта зона соответствует 23 и 24 индексам), так как это наилучшим образом позволяет нормализовать точки нижних конечностей. Несмотря на то, что в базу данных записывается весь скелет человека, именно движение нижних конечностей является определяющим при определении скорости перемещения человека. Таким образом, в качестве дополнительного преобразования, направленного на повышение точности работы нейросетевых алгоритмов, может выступать фильтрация точек модели тела человека: среди 33 точек рационально выбрать только шесть, соответствующие бедрам, коленям и стопам человека. Тогда получим следующие преобразования:

$$U_t = \left\{ \langle x_{t,i}, y_{t,i} \rangle \mid i = 0..32 \right\} \rightarrow \left\{ \langle x_{t,i} - x_t^n, y_{t,i} - y_t^n \rangle \mid i = 23..28 \right\},$$

$$x_t^n = \frac{(x_{t,23} + x_{t,24})}{2}, y_t^n = \frac{(y_{t,23} + y_{t,24})}{2}. \quad (3)$$

Сокращенный и нормированный объем данных позволит исключить влияние положения человека в кадре и движений его верхних конечностей, которые не всегда соответствуют процессу перемещения.

В рамках представленного алгоритма также рассмотрим ситуацию, при которой размеры объекты наблюдения могут изменяться в процессе движения, например, когда человек удаляется или приближается к камере, что может привести к существенной разнице в величине входных данных. Поэтому для таких случаев требуется выполнить нормирование модели тела человека относительно его роста. Такое преобразование позволит игнорировать размер распознанной в кадре модели тела человека. Тогда введем следующие обозначения:

$x_{\min} = \min(\{x_{t,i} \mid x_{t,i} \in U_t\})$  – минимальное значение по оси X среди всех точек тела человека;

$x_{\max} = \max(\{x_{t,i} \mid x_{t,i} \in U_t\})$  – максимальное значение по оси X среди всех точек тела человека;

$y_{\min} = \min(\{y_{t,i} \mid y_{t,i} \in U_t\})$  – минимальное значение по оси Y среди всех точек тела человека;

$y_{\max} = \max(\{y_{t,i} \mid y_{t,i} \in U_t\})$  – максимальное значение по оси Y среди всех точек тела человека;

Тогда:

$$U_t = \left\{ \left\langle x_{t,i}, y_{t,i} \right\rangle \right\} \rightarrow \left\{ \left\langle \frac{x_{t,i} - x_{\min}}{m_{xt}}, \frac{y_{t,i} - y_{\min}}{m_{yt}} \right\rangle \right\}, \quad (4)$$

$$m_{xt} = x_{\max} - x_{\min},$$

$$m_{yt} = y_{\max} - y_{\min},$$

$$i = 0..32.$$

Таким образом, дополнительные преобразования (3) и (4) позволяет повысить точность работы нейронных сетей за счет минимизации влияния положения человека в кадре и его текущих размеров.

**3. Сравнительный анализ нейросетевых технологий для прогнозирования движений человека.** Для решения поставленных задач (1) и (2) осуществлено сравнение различных алгоритмов машинного обучения и моделей нейронных сетей с целью выявления оптимального подхода, обеспечивающего наибольшую точность. Сводная таблица итоговых оптимальных характеристик алгоритмов и моделей представлена в таблице 1.

Таблица 1. Итоговые характеристики алгоритмов машинного обучения

Модель	Описание
DecisionTreeRegressor (DTR)	Стандартный decision tree regressor с параметром $\max\_depth = 5$
LinearRegression (LR)	Стандартная линейная регрессия с параметрами по умолчанию
Random Forest Regressor (RFR)	Стандартный регрессор случайного леса с параметрами $n\_estimators = 10, \max\_depth = 5$
Сверточная нейронная сеть (CNN)	Многослойная сверточная сеть из 4 пар слоев Conv1D + BatchNormalization (количество фильтров – от 32 до 256), слоем GlobalAvgPool2D и Dense
Рекуррентные нейронные сети (LSTM)	Многослойная нейронная сеть с 2 скрытыми Dense слоями из 100 и 200 нейронов, 2 LSTM слоя (20 нейронов)
Многослойные нейронные сети (NN)	Многослойная нейронная сеть с входом из скрытого Dense слоя из 200 нейронов с функцией активации ReLU и Dropout слоем (20% прореживание)

Архитектуры DTR, LR, RFR реализуется с использованием библиотеки scikit-learn [28], а CNN, LSTM и NN – на основе фреймворка Keras [29].

Каждая из архитектур использовалась для решения задач 1 и 2. Результаты для контрольной выборки, не участвующей в обучении или тестировании алгоритмов, представлены в таблице 2. Для первой задачи проведено 4 теста, в двух из которых прогнозируемые данные не используются ( $W = 0$ ). В экспериментах рассматривается две величины параметров  $Q$  и  $W$ : 10 и 15 кадров, что соответствует последовательностям движений длительностью 0,4 и 0,6 секунд соответственно. Параметры выбраны в соответствии со следующим расчетом:

Величина запаздывания  $\Omega = 0,5$  секунд.

Частота кадров камеры  $FPS = 25$ .

Время нейросетевой обработки  $\psi = 0,05$  секунд.

Тогда  $W = (0,5 + 0,05 + 0,04) \cdot 25 = 14,75$ , с округлением  $W = 15$ .

Значения  $Q = 10$  и  $W = 10$  взяты для сравнения точности прогнозирования, а также оценки работоспособности бесстрессового алгоритма с частичной компенсацией запаздывания.

Таблица 2. Сравнение алгоритмов машинного обучения по метрике MAE

Модель	Задача определения скорости (1)				Задача прогнозирования (2)	
	Q=10, W=0	Q=10, W=10	Q=15, W=0	Q=15, W=15	Q=10, W=10	Q=15, W=15
DTR	0,101	0,213	0,095	0,2	0,0033	0,0037
LR	0,496	0,489	0,489	1,47e+10	0,0034	0,0035
RFR	0,1	0,158	0,096	0,163	<b>0,0021</b>	<b>0,0022</b>
CNN	0,164	0,183	0,163	0,177	0,0101	0,0103
LSTM	0,196	0,216	0,171	0,198	0,0064	0,0069
NN	<b>0,084</b>	<b>0,133</b>	<b>0,081</b>	<b>0,129</b>	0,0044	0,0048

Для реализации бесстрессового алгоритма и проведения сравнительных испытаний использовались следующие библиотеки Python: OpenCV – для обработки видеопотока, MediaPipe для распознавания скелета человека в кадре, Keras – для формирования, обучения и применения моделей нейронных сетей, scikit-learn – для работы с алгоритмами машинного обучения. После сбора данных использовалось дополнительное преобразование (3) для нормирования координат нижних конечностей.

Общий объем собранных данных о перемещении человека с различной скоростью на активной беговой платформе составил 3495 записей для обучения и тестирования, 548 записей – для контрольной валидации. Исходные данные для обучения моделей машинного обучения сформированы в виртуальной среде Unity, что позволило фиксировать скорость и положение человека в произвольные моменты времени. Для оценки качества моделей использовалась метрика среднего абсолютного отклонения (MAE) между реальной скоростью и прогнозом. Для второй задачи было вычислено отклонение по каждой отслеживаемой точке модели тела человека, после чего определено MAE.

В первой задаче получено, что при определении скорости на основе только текущих значений положений за 10 или 15 кадров наилучшие показатели имеет модель NN. При добавлении прогнозов в размере 10 и 15 кадров наименьшую погрешность также имеет модель NN. Ближайшая по точности модель – RFR. Модель LR при  $Q = 15$ ,  $W = 15$  не смогла корректно аппроксимировать скорость, показывая слишком высокую погрешность, непригодную для использования (ошибка достигает  $1,468e+10$ ).

Решение задачи (2) обеспечивается практически всеми алгоритмами, наиболее эффективной является модель RFR. Результат применения модели RFR для прогнозирования положения и NN – для определения скорости представлен на рисунке 3.

Реальная скорость (текущая): 0.99	Прогноз. скорость (при $Q=10$ , $W=0$ ): 1.08
	Прогноз. скорость (при $Q=15$ , $W=0$ ): 1.03
Реальная скорость (при $W=10$ ): 1.11	Прогноз. скорость (при $Q=10$ , $W=10$ ): 1.22
Реальная скорость (при $W=15$ ): 0.65	Прогноз. скорость (при $Q=15$ , $W=15$ ): 0.71

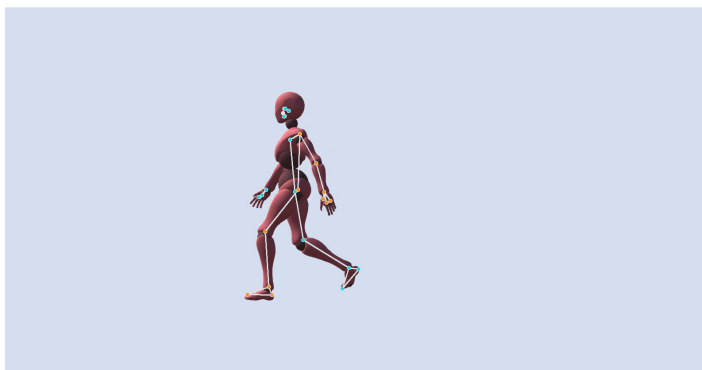


Рис. 3. Применение обученных моделей для прогнозирования скорости (пример 1, ходьба с последующим замедлением)

В эксперименте (рисунок 3) используется видео, полученное при захвате виртуальной сцены, в котором движется цифровая модель человека. Наблюдается определенный рост погрешности при использовании прогноза на 10 или 15 кадров, но это позволяет снизить запаздывание на 0,4 или 0,6 секунд соответственно. Оценивая погрешность в процессе движения, например, при скорости в 1 м/с, получаем достаточно высокую точность прогнозирования, причем, учитывается возможное торможение человека в процессе ходьбы (на рисунке 4 это событие началось через 15 кадров и достаточно точно было спрогнозировано). Таким образом, точность прогнозирования скорости с использованием RFR и NN составила от 91-96% (при  $W=0$ ) до 90,8% (при  $W=15$ ), что можно считать допустимым результатом.

Далее проведена апробация разработанного алгоритма в условиях равномерной ходьбы человека параллельно камере со средней скоростью в 1-1,5 м/с. Результаты прогнозирования скорости для реальных условий представлены на рисунке 4, где отражено две стадии ходьбы, полученные результаты соответствуют ожидаемому диапазону в 1-1,5 м/с. Кроме того на правом примере в момент остановки человека нейронная сеть также корректно отражает небольшую скорость (при  $Q = 0$ ,  $W = 0$ ).



Рис. 4. Применение обученных моделей для прогнозирования скорости (обычная ходьба)

Полученный результат позволяет сделать вывод о том, что нейросетевые алгоритмы прогнозирования могут использоваться для

определения скорости человека с точностью выше 90% на интервале прогноза до 0,6 секунд.

**4. Апробация разработанного алгоритма в системе управления беговой платформой.** Далее разработанное программное обеспечение используется непосредственно для управления беговой платформой. Корректирующие коэффициенты были установлены на следующие значения:  $k_1 = 0,3$  ,  $k_2 = 0,2$  ,  $k_3 = 0,5$  , таким образом, наибольший приоритет при расчете итоговой скорости отдается прогнозу  $s_p$  . Из-за необходимости обеспечения безопасности пользователя процесс тестирования был разделен на 3 этапа.

Первый этап реализован следующим образом: беговая платформа движется с фиксированной скоростью 1 м/с, система управления на основе компьютерного зрения осуществляет прогноз скорости, позволяя тем самым оценить величину отклонения. Фрагмент первого этапа представлен на рисунке 5. За счет учета всех составляющих (текущее положение, текущая и прогнозируемая скорость) итоговый расчет показывает достаточно близкие значения к реальной скорости. Здесь и далее выбраны значения  $Q = 10$  и  $W = 10$  , при которых расчет итоговой скорости показывает более корректные и точные значения, позволяя компенсировать 0,4 секунды запаздывания.

Прогнозируемая скорость:  
при  $Q=10$ ,  $W=0$ : 1.04  
при  $Q=10$ ,  $W=10$ : 0.87

Итоговый расчет: 1.02  
при  $Q=15$ ,  $W=0$ : 1.14  
при  $Q=15$ ,  $W=15$ : 0.31

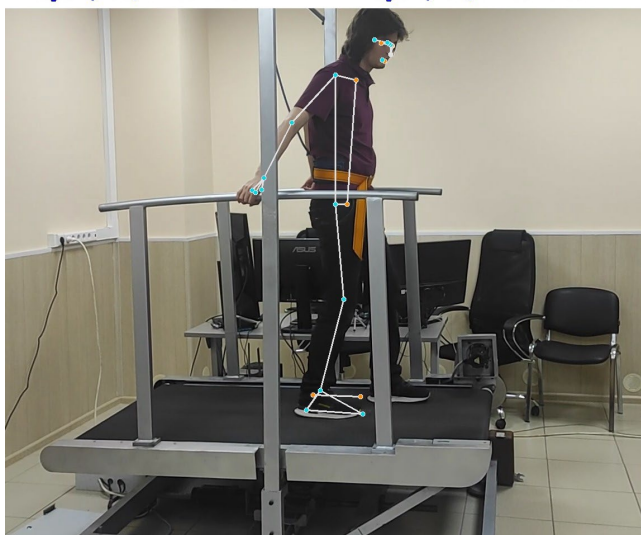




Рис. 5. Применение алгоритма для прогнозирования скорости на беговой платформе (первый этап тестирования)

На втором этапе тестирования беговая платформа была запущена в режиме адаптивного управления скоростью по трекерам виртуальной реальности, закрепленным на теле человека. Параллельно выполнялась запись прогноза скорости человека на основе разработанного алгоритма (без влияния на беговую платформу). Сравнение двух подходов представлено на рисунке 6. Необходимо отметить, что между двумя алгоритмами существует определенная разница, с другой стороны, предложенный подход на основе компьютерного зрения в целом совпадает по поведению с существующим, обрабатывает замедление, ускорение и остановку пользователя. Этот этап позволил проверить адекватность определения скорости при различных сценариях поведения пользователя, что необходимо перед переходом к финальному этапу тестирования.

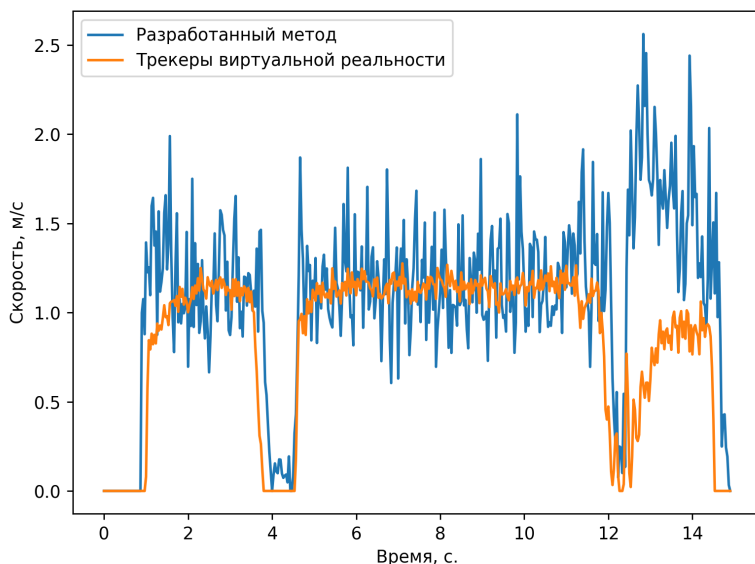


Рис. 6. Сравнение алгоритмов: разработанного и на основе трекеров виртуальной реальности (второй этап тестирования)

На третьем этапе тестирования (рисунок 7) для управления беговой платформой использовалось только разработанное программное обеспечение. График отражает скорость беговой платформы и положение человека на ней.

Необходимо отметить плавность и небольшую амплитуду изменения положения человека на платформе, отсутствуют моменты резкого изменения положения: среднее значение составило 0.15 метра при среднем отклонении 0.07 метра, что обеспечивает высокий комфорт пользователя при движении по платформе. Платформа также корректно останавливается при снижении скорости пользователя, причем, может осуществлять этот процесс не только на краю платформы, но и ближе к центру за счет получения нулевой скорости от нейронных сетей.

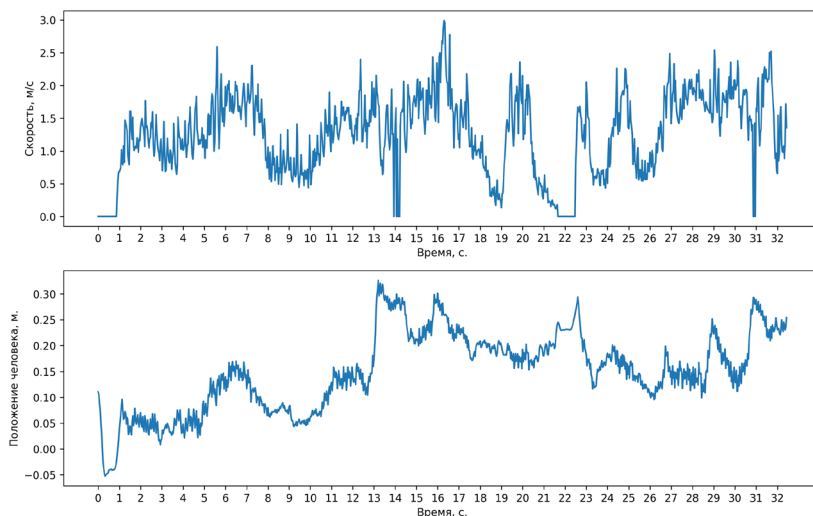


Рис. 7. Оценка качества работы разработанного алгоритма при управлении беговой платформой (третий этап тестирования)

Таким образом, разработанное программное обеспечение на основе бесстрессового алгоритма успешно интегрировано в систему управления беговой платформой. Разработанный алгоритм на основе комбинирования двух нейронных сетей и компьютерного зрения позволяет прогнозировать скорость человека в интервале от 10 до 15 кадров с высокой точностью (более 90%). Преимущество и новизна предлагаемого решения заключается в анализе текущего положения человека в кадре, определении текущей скорости и прогнозировании ее изменения, что в совокупности позволяет получить более точную оценку скорости, чем каждая из составляющих по отдельности. По сравнению с существующим подходом на основе трекеров виртуальной реальности предложенное решение обеспечивает

сопоставимый или больший комфорт пользователя (за счет уменьшения среднего отклонения положения человека на платформе и прогнозирования его действий), а также позволяет не использовать чувствительные к помехам датчики, требующие закрепления на теле человека.

**5. Заключение.** Поставлена задача прогнозирования скорости человека с использованием нейросетевых технологий и компьютерного зрения для минимизации запаздывания в системах управления беговыми платформами. Для ее решения разработан бесстрессовый алгоритм управления платформой, основанный на применении нейросетевых технологий для прогнозирования скорости и положения тела человека. На вход алгоритма поступает информация о положении точек тела человека, полученная при помощи технологий компьютерного зрения. Алгоритм объединяет три составляющие для получения итоговой скорости беговой платформы: анализ текущего положения человека в кадре, определение текущей скорости человека по набору последних  $Q$  кадров и прогнозирование скорости человека в интервале следующих  $W$  кадров.

В результате проведенных исследований был формализован алгоритм, включающий решение двух задач: определение положения тела человека через  $W$  количество кадров, а также определение скорости его движения по  $Q$  количеству кадров. В результате решения этих задач был снижен эффект запаздывания системы управления. В процессе исследования был проведен сбор данных о движениях человека с использованием компьютерного зрения и библиотеки MediaPipe, после чего осуществлено сравнение различных моделей нейронных сетей и алгоритмов машинного обучения для определения наиболее точного решения поставленных задач. Наилучшую точность при прогнозировании положения показал алгоритм RandomForestRegressor, а при определении скорости – многослойные нейронные сети. Получено, что при прогнозе положения тела человека в интервале длиной 15 кадров (приблизительно 0,6 секунд), скорость человека определяется с точностью более 90%. Проведены экспериментальные исследования по интеграции разработанного программного обеспечения в системы управления беговыми платформами и сравнению его с существующими решениями на основе трекеров виртуальной реальности. Из результатов экспериментов можно сделать вывод о дальнейших направлениях исследований: расширение обучающего датасета, необходимость удаления объектов окружения, перекрывающих человека (либо учет таких помех при обучении

моделей), масштабирование модели тела человека, что улучшит качество работы моделей в реальных условиях. Полученные результаты могут использоваться при разработке человеко-машинных систем, функционирующих на основе компьютерного зрения и анализа действий пользователя с учетом минимизации запаздывания системы и наличия ограничений на применение тактильных датчиков.

### Литература

1. Сиухин А.А., Карпушкин С.В. Определение шаблона движения пользователя на тренажере // Вестник Воронежского государственного технического университета. 2022. Т. 18. № 5. С. 48–55.
2. Kempfski K.M., Ray N.T., Knarr B.A., Higginson J.S. Dynamic structure of variability in joint angles and center of mass position during user-driven treadmill walking // *Gait & posture*. 2019. vol. 71. pp. 241–244.
3. Donlin M.C., Ray N.T., Higginson J.S. User-driven treadmill walking promotes healthy step width after stroke // *Gait & posture*. 2021. vol. 86. pp. 256–259.
4. Wehden L.O., Reer F., Zanzik R., Tang W.Y., Quandt T. The slippery path to total presence: How omnidirectional virtual reality treadmills influence the gaming experience // *Media and Communication*. 2021. vol. 9. no. 1. pp. 5–16.
5. Nath A.S. Eustress and distress—a stimulant and deterrent to health and performance // *Journal of Research Administration*. 2023. vol. 5. no. 2. pp. 5106–5121.
6. Hedjazi N., Benali A., Bouzit M., Dibi Z. An omnidirectional platform design: application to posture analysis // XIV Mediterranean Conference on Medical and Biological Engineering and Computing (IFMBE'2016). 2016. pp. 602–607.
7. Lichtenstein L., Barabas J., Woods R.L., Peli E.A. feedback-controlled interface for treadmill locomotion in virtual environments // *ACM Transactions on Applied Perception (TAP)*. 2007. vol. 4. no. 1.
8. De Luca A., Mattone R., Giordano P.R., Ulbrich H., Schwaiger M., Van den Bergh M., Koller-Meier E., Van Gool L. Motion control of the cybercarpet platform // *IEEE Transactions on Control Systems Technology*. 2012. pp. 410–427. DOI: 10.1109/TCST.2012.2185051.
9. Pyo S.H., Lee H., Yoon J. Development of a novel omnidirectional treadmill-based locomotion interface device with running capability // *Applied Sciences*. 2021. vol. 11(9). no. 4223.
10. Pyo S., Lee H., Yoon J. A Sensitive and Accurate Walking Speed Prediction Method Using Ankle Torque Estimation for a User-Driven Treadmill Interface // *IEEE Access*. 2022. vol. 10. pp. 102440–102450.
11. Lam W.W., Tang Y.M., Fong K.N. A systematic review of the applications of markerless motion capture (MMC) technology for clinical measurement in rehabilitation // *Journal of NeuroEngineering and Rehabilitation*. 2023. vol. 20. no. 1. pp. 1–26.
12. Souman J.L., Giordano P.R., Frissen I., Luca A.D., Ernst M.O. Making virtual walking real: Perceptual evaluation of a new treadmill control algorithm // *ACM Transactions on Applied Perception (TAP)*. 2010. vol. 7. no. 2. pp. 1–14.
13. Kikuchi T., Sakai K., Ishiya K. Gait Analysis with Automatic Speed-Controlled Treadmill // *Journal of Robotics and Mechatronics*. 2015. vol. 27. no. 5. pp. 528–534.
14. Auralius M., Yoon J.W. An automatic speed control system of a treadmill with ultrasonic sensors // *Journal of Institute of Control, Robotics and Systems*. 2011. vol. 17. no. 5. pp. 505–511.

15. Borges M., Symington A., Coltin B., Smith T., Ventura, R. Analysis and accuracy improvement // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2018). 2018. pp. 2610–2615.
16. Obukhov A., Dedov D., Volkov A., Teselkin D. Modeling of Nonlinear Dynamic Processes of Human Movement in Virtual Reality Based on Digital Shadows // Computation. 2023. vol. 11(5). no. 85.
17. Obukhov A., Volkov A., Pchelintsev A., Nazarova A., Teselkin D., Surkova E., Fedorchuk I. Examination of the Accuracy of Movement Tracking Systems for Monitoring Exercise for Musculoskeletal Rehabilitation // Sensors. 2023. vol. 23(19). no. 8058.
18. Chung J.L., Ong L.Y., Leow M.C. Comparative analysis of skeleton-based human pose estimation // Future Internet. 2022. vol. 14(12). no. 380.
19. Singhal R., Modi H., Srihari S., Gandhi A., Prakash C.O., Eswaran S. Body Posture Correction and Hand Gesture Detection Using Federated Learning and Mediapipe // 2nd International Conference for Innovation in Technology (INOCON'2023). 2023. pp. 1–6.
20. Obukhov A.D., Krasnyanskiy M.N., Dedov D.L., Vostrikova V.V., Teselkin D.V., Surkova E.O. Control of adaptive running platform based on machine vision technologies and neural networks // Neural Computing and Applications. 2022. vol. 34. no. 15. pp. 12919–12946.
21. Обухов А.Д., Назарова А.О. Метод управления на основе технологий компьютерного зрения и машинного обучения для адаптивных систем // Мехатроника, автоматизация, управление. 2023. Т. 24. № 1. С. 14–23.
22. Surkova E., Teselkin D., Nazarova A., Arkhipov A. Structural and functional models of adaptive motion control system in virtual reality // AIP Conference Proceedings. 2023. vol. 2910. no. 1.
23. Su B., Smith C., Gutierrez Fawerik E. Gait phase recognition using deep convolutional neural network with inertial measurement units // Biosensors. 2020. vol. 10(9). no. 109.
24. Li Y., Zhang S., Wang Z., Yang S., Yang W., Xia S.T., Zhou E. Tokenpose: Learning keypoint tokens for human pose estimation // Proceedings of the IEEE/CVF International conference on computer vision (ICCVW'2021). 2021. pp. 11313–11322.
25. Yacchirema D., de Puga J.S., Palau C., Esteve M. Fall detection system for elderly people using IoT and ensemble machine learning algorithm // Personal and Ubiquitous Computing. 2019. vol. 23. pp. 801–817.
26. Bartol K., Bojanic D., Petkovic T., Peharec S., Pribanic T. Linear regression vs. deep learning: A simple yet effective baseline for human body measurement // Sensors. 2022. vol. 22(5). no. 1885.
27. Turgeon S., Lanovaz M.J. Tutorial: Applying machine learning in behavioral research // Perspectives on Behavior Science. 2020. vol. 43. no. 4. pp. 697–723.
28. Gupta A., Gupta K., Gupta K., Gupta K. Human Activity Recognition Using Pose Estimation and Machine Learning Algorithm // ISIC. 2021. vol. 21. pp. 25–27.
29. Pham H.H., Salmane H., Khoudour L., Crouzil A., Velastin S.A., Zegers P. A unified deep framework for joint 3d pose estimation and action recognition from a single RGB camera // Sensors. 2020. vol. 20(7). no. 1825.

**Обухов Артем Дмитриевич** — д-р техн. наук, доцент, руководитель лаборатории, лаборатория VR тренажеров, управление фундаментальных и прикладных исследований, Тамбовский государственный технический университет. Область научных интересов: адаптивные информационные системы, структурно-параметрический синтез, нейронные сети, машинное обучение. Число научных

публикаций — 248. obuhov.art@gmail.com; улица Советская, 106, 392000, Тамбов, Россия; р.т.: +7(915)867-6915.

**Дедов Денис Леонидович** — канд. техн. наук, старший научный сотрудник, управление фундаментальных и прикладных исследований, Тамбовский государственный технический университет. Область научных интересов: виртуальные тренажерные комплексы, обучение персонала, информационные технологии, адаптивные информационные системы. Число научных публикаций — 128. hammer68@mail.ru; улица Советская, 106, 392000, Тамбов, Россия; р.т.: +7(915)672-9838.

**Теселкин Даниил Вячеславович** — техник-программист, управление фундаментальных и прикладных исследований, Тамбовский государственный технический университет. Область научных интересов: разработка программного обеспечения, виртуальная реальность, компьютерное зрение, анализ данных. Число научных публикаций — 26. dteselk@mail.ru; улица Советская, 106, 392000, Тамбов, Россия; р.т.: +7(920)482-7613.

**Волков Андрей Андреевич** — младший научный сотрудник, управление фундаментальных и прикладных исследований, Тамбовский государственный технический университет. Область научных интересов: информационные технологии, архитектуры информационных систем, распределенные системы управления. Число научных публикаций — 39. didim@eclabs.ru; улица Советская, 106, 392000, Тамбов, Россия; р.т.: +7(953)703-0619.

**Назарова Александра Олеговна** — техник-программист, управление фундаментальных и прикладных исследований, Тамбовский государственный технический университет. Область научных интересов: информационные технологии, разработка программного обеспечения, виртуальная реальность, машинное обучение, анализ данных. Число научных публикаций — 29. nazarova.al.ol@yandex.ru; улица Советская, 106, 392000, Тамбов, Россия; р.т.: +7(920)496-3910.

**Поддержка исследований.** Исследование выполнено за счет гранта Российского научного фонда № 22-71-10057, <https://rscf.ru/project/22-71-10057/>.

A. OBUKHOV, D. DEDOV, D. TESELKIN, A. VOLKOV, A. NAZAROVA  
**DEVELOPMENT OF A STRESS-FREE ALGORITHM FOR  
CONTROL OF RUNNING PLATFORMS BASED ON NEURAL  
NETWORK TECHNOLOGIES**

*Obukhov A., Dedov D., Teselkin D., Volkov A., Nazarova A.* **Development of a Stress-Free Algorithm for Control of Running Platforms Based on Neural Network Technologies.**

**Abstract.** The article discusses the task of predicting human speed using neural network technologies and computer vision to minimize lags in treadmill control systems, which pose a health risk to the user. To solve this problem, a stress-free algorithm has been developed, including: predicting the position and speed of the user on the treadmill; calculating the treadmill speed based on the analysis of the user's position and movement characteristics; data collection and processing schemes for training neural network methods; and determining the necessary number of predicted frames to eliminate lags. The scientific novelty of the research lies in the development of a treadmill control algorithm that combines: computer vision technologies for recognizing the user's body model on the platform; neural networks; and machine learning methods to determine the final human speed based on combining data on the person's position in the frame and the current and predicted speed of the person. The proposed algorithm is implemented using Python libraries, and its validation was conducted during experimental studies analyzing the preceding 10 and 15 frames to predict the next 10 and 15 frames. Comparing machine learning algorithms (linear regression, decision tree, random forest, multilayer, convolutional, and recurrent neural networks) at different lengths of analyzed and predicted frames, the RandomForestRegressor algorithm showed the best accuracy in predicting position, while dense multilayer neural networks performed best in determining current speed. Experimental research has been conducted on applying the developed algorithm and models to determine human speed (achieving accuracy when forecasting in the range of 10-15 frames) as well as integrating them into treadmill control systems. Trials have shown the effectiveness of the proposed approach and the correctness of system operation under real conditions. The developed algorithm allows for not using noise-sensitive sensors that require attachment to the user's body but rather forecasting user actions through analyzing all points of the person's body to reduce lags in various human-machine systems.

**Keywords:** running platforms, neural network technologies, stress-free control algorithm, machine learning.

## References

1. Siukhin A.A., Karpushkin S.V. [Determining the user's movement pattern on a simulator]. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta – Bulletin of Voronezh state technical University*. 2022. vol. 18. no. 5. pp. 48–55. (In Russ.).
2. Kempfski K.M., Ray N.T., Knarr B.A., Higginson J.S. Dynamic structure of variability in joint angles and center of mass position during user-driven treadmill walking. *Gait & posture*. 2019. vol. 71. pp. 241–244.
3. Donlin M.C., Ray N.T., Higginson J.S. User-driven treadmill walking promotes healthy step width after stroke. *Gait & posture*. 2021. vol. 86. pp. 256–259.
4. Wehden L.O., Reer F., Janzik R., Tang W.Y., Quandt T. The slippery path to total presence: How omnidirectional virtual reality treadmills influence the gaming experience. *Media and Communication*. 2021. vol. 9. no. 1. pp. 5–16.

5. Nath A.S. Eustress and distress-a stimulant and deterrent to health and performance. *Journal of Research Administration*. 2023. vol. 5. no. 2. pp. 5106–5121.
6. Hedjazi N., Benali A., Bouzit M., Dibi Z. An omnidirectional platform design: application to posture analysis. *XIV Mediterranean Conference on Medical and Biological Engineering and Computing (IFMBE'2016)*. 2016. pp. 602–607.
7. Lichtenstein L., Barabas J., Woods R.L., Peli E.A. feedback-controlled interface for treadmill locomotion in virtual environments. *ACM Transactions on Applied Perception (TAP)*. 2007. vol. 4. no. 1.
8. De Luca A., Mattone R., Giordano P.R., Ulbrich H., Schwaiger M., Van den Bergh M., Koller-Meier E., Van Gool L. Motion control of the cybercarpet platform. *IEEE Transactions on Control Systems Technology*. 2012. pp. 410–427. DOI: 10.1109/TCST.2012.2185051.
9. Pyo S.H., Lee H., Yoon J. Development of a novel omnidirectional treadmill-based locomotion interface device with running capability. *Applied Sciences*. 2021. vol. 11(9). no. 4223.
10. Pyo S., Lee H., Yoon J. A Sensitive and Accurate Walking Speed Prediction Method Using Ankle Torque Estimation for a User-Driven Treadmill Interface. *IEEE Access*. 2022. vol. 10. pp. 102440–102450.
11. Lam W.W., Tang Y.M., Fong K.N. A systematic review of the applications of markerless motion capture (MMC) technology for clinical measurement in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*. 2023. vol. 20. no. 1. pp. 1–26.
12. Souman J.L., Giordano P.R., Frissen I., Luca A.D., Ernst M.O. Making virtual walking real: Perceptual evaluation of a new treadmill control algorithm. *ACM Transactions on Applied Perception (TAP)*. 2010. vol. 7. no. 2. pp. 1–14.
13. Kikuchi T., Sakai K., Ishiya K. Gait Analysis with Automatic Speed-Controlled Treadmill. *Journal of Robotics and Mechatronics*. 2015. vol. 27. no. 5. pp. 528–534.
14. Auralius M., Yoon J.W. An automatic speed control system of a treadmill with ultrasonic sensors. *Journal of Institute of Control, Robotics and Systems*. 2011. vol. 17. no. 5. pp. 505–511.
15. Borges M., Symington A., Coltin B., Smith T., Ventura, R. Analysis and accuracy improvement. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2018)*. 2018. pp. 2610–2615.
16. Obukhov A., Dedov D., Volkov A., Teselkin D. Modeling of Nonlinear Dynamic Processes of Human Movement in Virtual Reality Based on Digital Shadows. *Computation*. 2023. vol. 11(5). no. 85.
17. Obukhov A., Volkov A., Pchelintsev A., Nazarova A., Teselkin D., Surkova E., Fedorchuk I. Examination of the Accuracy of Movement Tracking Systems for Monitoring Exercise for Musculoskeletal Rehabilitation. *Sensors*. 2023. vol. 23(19). no. 8058.
18. Chung J.L., Ong L.Y., Leow M.C. Comparative analysis of skeleton-based human pose estimation. *Future Internet*. 2022. vol. 14(12). no. 380.
19. Singhal R., Modi H., Srihari S., Gandhi A., Prakash C.O., Eswaran S. Body Posture Correction and Hand Gesture Detection Using Federated Learning and Mediapipe. *2nd International Conference for Innovation in Technology (INOCON'2023)*. 2023. pp. 1–6.
20. Obukhov A.D., Krasnyanskiy M.N., Dedov D.L., Vostrikova V.V., Teselkin D.V., Surkova E.O. Control of adaptive running platform based on machine vision technologies and neural networks. *Neural Computing and Applications*. 2022. vol. 34. no. 15. pp. 12919–12946.
21. Obukhov A.D., Nazarova A.O. [A control method based on computer vision and machine learning technologies for adaptive systems]. *Mekhatronika, avtomatizaciya,*



- upravlenie – Mechatronics, automation, control. 2023. vol. 24. no. 1. pp. 14–23. (In Russ.).
22. Surkova E., Teselkin D., Nazarova A., Arkhipov A. Structural and functional models of adaptive motion control system in virtual reality. AIP Conference Proceedings. 2023. vol. 2910. no. 1.
  23. Su B., Smith C., Gutierrez Farewik E. Gait phase recognition using deep convolutional neural network with inertial measurement units. Biosensors. 2020. vol. 10(9). no. 109.
  24. Li Y., Zhang S., Wang Z., Yang S., Yang W., Xia S.T., Zhou E. Tokenpose: Learning keypoint tokens for human pose estimation. Proceedings of the IEEE/CVF International conference on computer vision (ICCVW'2021). 2021. pp. 11313–11322.
  25. Yacchirema D., de Puga J.S., Palau C., Esteve M. Fall detection system for elderly people using IoT and ensemble machine learning algorithm. Personal and Ubiquitous Computing. 2019. vol. 23. pp. 801–817.
  26. Bartol K., Bojanic D., Petkovic T., Peharec S., Pribanic T. Linear regression vs. deep learning: A simple yet effective baseline for human body measurement. Sensors. 2022. vol. 22(5). no. 1885.
  27. Turgeon S., Lanovaz M.J. Tutorial: Applying machine learning in behavioral research. Perspectives on Behavior Science. 2020. vol. 43. no. 4. pp. 697–723.
  28. Gupta A., Gupta K., Gupta K., Gupta K. Human Activity Recognition Using Pose Estimation and Machine Learning Algorithm. ISIC. 2021. vol. 21. pp. 25–27.
  29. Pham H.H., Salmane H., Khoudour L., Crouzil A., Velastin S.A., Zegers P. A unified deep framework for joint 3d pose estimation and action recognition from a single RGB camera. Sensors. 2020. vol. 20(7). no. 1825.

**Obukhov Artem** — Ph.D., Dr.Sci., Associate Professor, Head of the laboratory, Department of fundamental and applied research, Tambov State Technical University. Research interests: adaptive information systems, structural-parametric synthesis, neural networks, machine learning. The number of publications — 248. obuhov.art@gmail.com; 106, Sovetskaya St., 392000, Tambov, Russia; office phone: +7(915)867-6915.

**Dedov Denis** — Ph.D., Senior researcher, Department of fundamental and applied research, Tambov State Technical University. Research interests: virtual training complexes, personnel training, information technologies, adaptive information systems. The number of publications — 128. hammer68@mail.ru; 106, Sovetskaya St., 392000, Tambov, Russia; office phone: +7(915)672-9838.

**Teselkin Daniil** — Programmer technician, Department of fundamental and applied research, Tambov State Technical University. Research interests: information technology, software development, virtual reality, computer vision, data analysis. The number of publications — 26. dteselk@mail.ru; 106, Sovetskaya St., 392000, Tambov, Russia; office phone: +7(920)482-7613.

**Volkov Andrey** — Junior researcher, Department of fundamental and applied research, Tambov State Technical University. Research interests: information technologies, information system architectures, distributed control systems. The number of publications — 39. didim@eclabs.ru; 106, Sovetskaya St., 392000, Tambov, Russia; office phone: +7(953)703-0619.

**Nazarova Alexandra** — Programmer technician, Department of fundamental and applied research, Tambov State Technical University. Research interests: information technology, software development, virtual reality, machine learning, data analysis. The number

of publications — 29. nazarova.al.ol@yandex.ru; 106, Sovetskaya St., 392000, Tambov, Russia; office phone: +7(920)496-3910.

**Acknowledgements.** The research was carried out at the expense of the grant of the Russian Science Foundation No. 22-71-10057, <https://rscf.ru/en/project/22-71-10057/>.

## Руководство для авторов

Взаимодействие автора с редакцией осуществляется через личный кабинет на сайте журнала «Информатика и автоматизация» <http://ia.spcras.ru/>. При регистрации авторам рекомендуется заполнить все предложенные поля данных. Подготовка статьи ведется с помощью текстовых редакторов MS Word 2007 и выше или LaTeX. Объем основного текста (до раздела Литература) - от 20 до 30 страниц включительно. Переносы запрещены. Номера страниц не проставляются. Основная часть текста статьи разбивается на разделы, среди которых являются обязательными: введение, хотя бы один «содержательный» раздел и заключение. Допускается также мотивированное содержанием и структурой материал а выделение подразделов. В основную часть опускается помещать рисунки, таблицы, листинги и формулы. Правила их оформления подробно рассмотрены на нашем сайте в разделе «Руководство для авторов».

## Author guidelines

Interaction between each potential author and the Editorial board is realized through the pesoal account on the website of the journal "Informatics and Automation" <http://ia.spcras.ru/>. At the registration the authors are requested to fill out all data fields in the proposed form. The submissions should be prepared using MS Word 2007, LaTeX. The text of the paper in the main part should not exceed 30 pages. Pages are not numbered; hyphenations are not allowed. Certain figures, tables, listings and formulas are allowed in the main section, and their typography is considered in more detail at the journal web.

---

Signed to print 28.05.2024. Passed for print 29.05.2024.

Printed in Publishing center GUAP.

Address: 67 litera A, B. Morskaya, St. Petersburg, 190000, Russia

---

Founder and Publisher: SPC RAS.

Address: 39 litera A, 14th Line V.O., St. Peterburg, 199178, Russia.

The journal is registered in the Federal Service for Supervision of Communications, Information Technology, and Mass Media, Registration Certificate (registration number) ПИ № ФС77-79228 dated September 25, 2020 Subscription Index П5513, Russian Post Catalog

---

Подписано к печати 28.05.2024. Дата выхода в свет 29.05.2024.

Формат 60×90 1/16. Усл. печ. л. 19,2. Заказ № 176. Тираж 300 экз., цена свободная.

Отпечатано в Редакционно-издательском центре ГУАП.

Адрес типографии: Б. Морская, д. 67, лит. А, г. Санкт-Петербург, 190000, Россия

---

Учредитель и издатель: СПб ФИЦ РАН.

Адрес учредителя и издателя: 14-я линия В.О., д. 39, лит. А, г. Санкт-Петербург, 199178, Россия

Журнал зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций, свидетельство о регистрации (регистрационный номер) ПИ № ФС77-79228 от 25 сентября 2020 г.

Подписной индекс П5513 по каталогу «Почта России»