Intellectual Technologies on Transport No 2

Интеллектуальные технологии на транспорте № 2

Санкт-Петербург
St. Petersburg
2022

Интеллектуальные технологии на транспорте
(сетевой электронный научный журнал)
№ 2, 2022

ISSN 2413-2527

Сетевой электронный научный журнал, свободно распространяемый через Интернет.
Публикуются статьи на русском и английском языках с результатами исследований
и практических достижений в области интеллектуальных технологий
и сопутствующих им научных исследований.

Журнал основан в 2015 году.

Учредитель и издатель

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I» (ФГБОУ ВО ПГУПС)

Главный редактор

Хомоненко А. Д., д.т.н., проф., С.-Петербург, РФ

Сопредседатели редакционного совета

Панычев А. Ю., ректор ПГУПС, С.-Петербург, РФ
Чаркин Е. И., зам. ген. директора по ИТ ОАО «РЖД», Москва, РФ

Редакционный совет

Ададуров С. Е., проф., Москва, РФ
Дудин А. Н., д.т.н., проф., БГУ, Минск, Беларусь
Корниенко А. А., проф., ПГУПС, С.-Петербург, РФ
Ковалец П., проф., Техн. ун-т, Варшава, Польша
Меркурьев Ю. А., проф., РТУ, Рига, Латвия
Нестеров В. М., проф., СПбГУ, С.-Петербург, РФ
Пустарнаков В. Ф., зам. ген. дир. «Газинформсервис», С.-Петербург, РФ
Титова Т. С., проф., проректор ПГУПС, С.-Петербург, РФ
Федоров А. Р., ген. дир. «Digital Design», С.-Петербург, РФ
Юсупов Р. М., проф., чл.-корр. РАН, С.-Петербург, РФ

Редакционная коллегия

Бубнов В. П., проф., С.-Петербург, РФ – заместитель главного редактора
Александрова Е. Б., проф., С.-Петербург, РФ
Атилла Элчи, проф., ун-т Аксарай, Турция
Басыров А. Г., проф., С.-Петербург, РФ
Безродный Б. Ф., проф., Москва, РФ
Благовещенская Е. А., проф., С.-Петербург, РФ
Булавский П. Е., д.т.н., доц., С.-Петербург, РФ
Василенко М. Н., проф., С.-Петербург, РФ
Глухов А. П., д.т.н., Москва, РФ
Гуда А. Н., проф., Ростов-на-Дону, РФ
Железняк В. К., проф., Новополоцк, Беларусь
Заборовский В. С., проф., С.-Петербург, РФ
Канаев А. К., проф., С.-Петербург, РФ
Котенко А. Г., д.т.н., доц., С.-Петербург, РФ
Куренков П. В., проф., Москва, РФ
Лецкий Э. К., проф., Москва, РФ
Макаренко С. И., д.т.н., доц., С.-Петербург, РФ
Мирзоев Т. А., асс. проф., Джорджия, США
Наседкин О. А., к.т.н., доц., С.-Петербург, РФ
Никитин А. Б., проф., С.-Петербург, РФ
Новиков Е. А., д.т.н., доц., С.-Петербург, РФ
Охтилев М. Ю., проф., С.-Петербург, РФ
Привалов А. А., проф., С.-Петербург, РФ
Соколов Б. В., проф., С.-Петербург, РФ
Таранцев А. А., проф., С.-Петербург, РФ
Утепбергенов И. Т., проф., Алматы, Казахстан
Филипченко С. А., к.т.н., доц., Москва, РФ
Фозилов Ш. Х., проф., Ташкент, Узбекистан
Хабаров В. И., проф., Новосибирск, РФ
Ходаковский В. А., проф., С.-Петербург, РФ
Ху Фу-Ниан, проф., Сюйчжоу, Китай
Чехонин К. А., проф., Хабаровск, РФ
Ялышев Ю. И., проф., Екатеринбург, РФ

Адрес редакции:
190031, Санкт-Петербург, Московский пр., 9
e-mail: itt-pgups@yandex.ru
Телефон: +7 (812) 457-86-06

Сетевое издание «Интеллектуальные технологии на транспорте (сетевой электронный научный журнал), Intellectual Technologies on Transport» зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.
Свидетельство Эл № ФС77–61707 от 07 мая 2015 г.

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ).

Периодичность выхода – 4 номера в год. Выпуски журнала доступны на сайте http://itt-pgups.ru.

Информация предназначена для детей старше 12 лет.

© Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I», 2022

Intellectual Technologies on Transport
Issue 2, 2022

ISSN 2413-2527

Network electronic scientific journal, open access. It publishes articles in Russian and English with the results of research and practical achievements in the field of intelligent technologies and associated research.

Founded in 2015.

Founder and Publisher
Federal State Educational Institution of Higher Education
«Emperor Alexander I St. Petersburg State Transport University»

Editor-in-Chief
Khomonenko A. D., Dr. Sc., Prof., St. Petersburg, Russia

Co-chairs of the Editorial Council
Panychev A. Y., rector of PSTU, St. Petersburg, Russia
Charkin E. I., CIO of JSC «Russian Railways», Moscow, Russia

Editorial Council Members

Adadurov S. E., Prof., Moscow, Russia
Dudin A. N., Prof., BSU, Minsk, Belarus
Kornienko A. A., Prof., PSTU, St. Petersburg, Russia
Kovalets P., Prof., Tech. University, Warsaw, Poland
Merkuryev Y. A., Prof., RTU, Academician of the Latvian Academy of Sciences, Riga, Latvia
Nesterov V. M., Prof., SPbSU, St. Petersburg, Russia
Pustarnakov V. F., Deputy CEO at «Gazinformservice» Ltd., St. Petersburg, Russia
Titova T. S., Prof., Vice-Rector, PSTU, St. Petersburg, Russia
Fedorov A. R., CEO at «Digital Design» Ltd., St. Petersburg, Russia
Yusupov R. M., Prof., Corr. Member of RAS, St. Petersburg, Russia

Editorial Board Members

Bubnov V. P., Prof., St. Petersburg, Russia – Deputy Editor-in-Chief
Aleksandrova E. B., Prof., St. Petersburg, Russia
Atilla Elci, Prof., Aksaray University, Turkey
Basyrov A. G., Prof., St. Petersburg, Russia
Bezrodny B. F., Prof., Moscow, Russia
Blagoveshchenskaya E. A., Prof., St. Petersburg, Russia
Bulavsky P. E., Dr. Sc., As. Prof., St. Petersburg, Russia
Vasilenko M. N., Prof., St. Petersburg, Russia
Glukhov A. P., Dr. Sc., St. Petersburg, Russia
Guda A. N., Prof., Rostov-on-Don, Russia
Zheleznyak V. K., Prof., Novopolotsk, Belarus
Zaborovsky V. S., Prof., St. Petersburg, Russia
Kanaev A. K., Prof., St. Petersburg, Russia
Kotenko A. G., Dr. Sc., As. Prof., St. Petersburg, Russia
Kurenkov P. V., Prof., Moscow, Russia
Letsky E. K., Prof., Moscow, Russia
Makarenko S. I., Dr. Sc., As. Prof., St. Petersburg, Russia
Mirzoev T. A., As. Prof., Georgia, USA
Nasedkin O. A., As. Prof., St. Petersburg, Russia
Nikitin A. B., Prof., St. Petersburg, Russia
Novikov E. A., Dr. Sc., As. Prof., St. Petersburg, Russia
Okhtilev M. Y., Prof., St. Petersburg, Russia
Privalov A. A., Prof., St. Petersburg, Russia
Sokolov B. V., Prof., St. Petersburg, Russia
Tarantsev A. A., Prof., St. Petersburg, Russia
Utepbergenov I. T., Prof., Almaty, Kazakhstan
Filipchenko S. A., As. Prof., Moscow, Russia
Fozilov Sh. Kh., Prof., Tashkent, Uzbekistan
Khabarov V. I., Prof., Novosibirsk, Russia
Khodakovsky V. A., Prof., St. Petersburg, Russia
Hu Fu-Nian, Prof., Xuzhou, China
Chekhonin K. A., Prof., Khabarovsk, Russia
Yalyshev Y. I., Prof., Ekaterinburg, Russia

Editorial adress:
190031, St. Petersburg, Moskovsky ave., 9
e-mail: itt-pgups@yandex.ru
Phone: +7 812 457 86 06

The online journal «Intellectual Technologies on Transport» is registered by the Federal Service for Supervision of Communications, Information Technologies, and Mass Media.
El No. FS77-61707 Testimony from May 7, 2015.

The journal is registered in the Russian Science Citation Index (RSCI).

Frequency of release - 4 issues per year. Issues of the magazine are available at http://itt-pgups.ru

The content is for children over the age of 12.

© Federal State Budgetary Educational Institution of Higher Education
«Emperor Alexander I St. Petersburg State Transport University», 2022

Содержание

Смагин В. А., Якубович Г. Л.
Обобщение понятия ресурса надежности технической системы Н. М. Седякина до ресурса работоспособности живого организма ……...……..... 5

Моисеев В. И., Ксенофонтова В. А., Комарова Т. А.
Математическая модель процесса плавления высоковязкого нефтепродукта при выгрузке из котла вагона-цистерны …….....…..….. 10

Дурницын О. А.
Способ фильтрации данных GPS для цикла движения. Исследование топливной экономичности специального автомобиля …..……...…... 19

Дмитриева М. С., Забродин А. В.
Автоматизация документооборота и бизнес-процессов с помощью ECM-системы Docsvision ……. 25

Гончаренко В. А., Забузов В. С.
Методы построения кластерных космических систем на основе многоагентных технологий …..... 32

Брехт Э. А., Коншина В. Н.
Применение нейронной сети YOLO для распознавания дефектов …..…... 41

Сунь Ифань, Гильванов Р. Г.
Реализация анализа главных компонент с RHadoop ……..…... 48

Федотова А. И., Гильванов Р. Г.
Разработка кроссплатформенных приложений на языке Python и фреймворке Kivy ……...…... 53

Contents

Smagin V. A., Yakubovich G. L.
Generalization of N. M. Sedyakin`s Concept of the Reliability Resource of a Technical System to the Working Capacity Resource of a Living Organism ……..…..……..…. 5

Moiseev V. I., Ksenofontova V. A., Komarova T. A.
Mathematical Model of Melting Process of High Viscosity Oil Product Upon Discharge from of Tank Wagon ……..…. 10

Durnitsin O. A.
A Way to Filter GPS Data for a Driving Cycle. Investigation of the Fuel Efficiency of a Special Car …..... 19

Dmitrieva M. S., Zabrodin A. V.
Automation of Document-Flow and Business Processes Using the ECM-System Docsvision ……...…. 25

Goncharenko V. A., Zabuzov V. S.
Methods for Constructing Cluster Space Systems Based on Multi-Agent Technologies ……..…...…… 32

Breht E. A., Konshina V. N.
Application of YOLO Neural Network for Defect Recognition ……....……...…...… 41

Sun Yifan, Gilvanov R. G.
Implementing Principal Component Analysis with RHadoop ……..…..… 48

Fedotova A. I., Gilvanov R. G.
Development of Cross-Platform Applications with Python and Kivy Framework ……..…. 53

DOI: 10.24412/2413-2527-2022-230-5-9

Обобщение понятия ресурса надежности технической системы Н. М. Седякина до ресурса работоспособности живого организма

засл. деятель науки РФ д.т.н., проф. В. А. Смагин, Г. Л. Якубович
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия
va_smagin@mail.ru, g.l.yakubovich@gmail.com

Аннотация. **Предложено количественное обобщение понятия профессора Н. М. Седякина о ресурсе надежности объекта до ресурса работоспособности человека. Первый этап связан с введением понятия ресурса надежности объекта как основного показателя надежности систем, который объект имеет и вырабатывает во время применения по назначению. Второй этап связан с явлением увеличения продолжительности действия объекта за счет устранения его возможных отказов и дефектов благодаря доработкам технического объекта и тестирования функционирования программного обеспечения объекта перед применением по назначению. На основе выделения двух этапов предложено количественное обобщение их частных показателей в один общий показатель — ресурс работоспособности объекта. Выполнено исследование свойств этого показателя и предложены рекомендации по его использованию и дальнейшему развитию в стратегиях жизненной деятельности человека с целью повышения его работоспособности.**

Ключевые слова: **ресурс надежности, вероятностная модель, ресурс работоспособности человека, жизненная стратегия, восполняемый ресурс.**

Введение

19 апреля 2022 года исполнилось 100 лет со дня рождения выдающегося ученого Николая Михайловича Седякина — доктора технических наук, профессора, начальника ведущей кафедры эксплуатации автоматизированных систем управления Военно-космической академии имени А. Ф. Можайского, известного ученого в теории случайных импульсных потоков, теории эксплуатации сложных систем и теории надежности. В 1965 году им сделан доклад на тему «Об одном физическом принципе теории надежности и его приложениях», а в 1966 году опубликована статья в «Известиях АН СССР. Техническая кибернетика» [1]. Сущность его работы сводится к тому, что им впервые сформулирован основной закон теории надежности, который гласит, что надежность системы зависит от величины выработанного ею ресурса надежности и не зависит от того, как он был выработан. Ресурс надежности количественно определялся в виде:

$$r(t,\varepsilon) = \int_0^t \lambda(z,\varepsilon)dz\,,$$

где $\lambda(t,\varepsilon)$ — интенсивность отказа системы, работающей течение времени t в условиях физического нагружения ε.

Закон многократно проверялся экспериментально на различных технических элементах, и было установлено, что он выполнялся при недостаточно сильных физических нагружениях. Но при сильных нагружениях его выполнение не наблюдалось. Тем не менее, на его основе был написан ряд научных работ докторского характера, и он использовался при ускоренных и форсированных испытаниях элементов и систем на надежность.

Автор данной статьи выполнил ряд научных исследований и опубликовал предложение с теоретическим обобщением закона [2]. Сущность этого предложения заключалась в необходимости учета при расчетах надежности перепадов величин нагружений и времени их наблюдения.

Если ранее вероятность безотказной работы объекта определялась по формуле

$$P(t,\varepsilon) = e^{-\int_0^t \lambda(z,\varepsilon)dz}\,, \qquad (1)$$

то после теоретического обобщения было рекомендовано вводить относительное условное нагружение, приводящее к необходимости учитывать, как в электротехнике, потенциал и напряжение, и выполнять расчеты надежности с применением не только интегральной формы расчета (1), но и с переходом к интегрально-дифференциальной форме. Поясним это на одном частном примере расчета величины интенсивности отказа объекта и тогда, когда нас интересует переход его от более жесткого режима работы к менее жесткому режиму [3]:

$$\lambda(t,\varepsilon_0) = \frac{\varepsilon_0}{\varepsilon_*}\lambda\left(\frac{\varepsilon_0}{\varepsilon_*}t,\varepsilon_9\right) - (\varepsilon_0-\varepsilon_*)\frac{\partial\lambda\left(\frac{\varepsilon_0}{\varepsilon_*}t,\varepsilon_*\right)}{\partial\varepsilon_*}\,.$$

Для других изменений режимов выражения можно найти в указанном источнике.

Надежность программного обеспечения и формула D. Musa

Если 60-е годы прошлого столетия были годами появления и становления теории надежности аппаратных средств, то 80-е годы были годами рождения и развития теории надежности программных средств. Конечно, доработки на технике были связаны с повышением надежности и готовности на первом этапе развития этой науки. Но они не были столь характерными, как на втором этапе развития, а именно периоде развития программной надежности. Поэтому первый характеризуется появлением понятия ресурса надежности и формулы Н. М. Седякина для него, а второй этап характеризуется появлением модели доработок в программном обеспечении в виде тестирования и формулы D. Musa для него [4].

В своей работе D. Musa показал, какого эффекта в надежности программы можно достичь, если до ее применения по назначению многократно подвергать тестированию с целью обнаружения и устранения дефектов в ней. Он предложил простейшую математическую модель расчета безошибочности программы с учетом ее предварительного тестирования до использования ее по назначению [4]. Если воспользоваться понятием ресурса надежности Н. М. Седякина, то в обобщенном виде можно записать выражение для вероятности ее безошибочного функционирования:

$$P(r,\rho) = e^{-r \times e^{-\rho}},$$

где r — обычный ресурс надежности Н. М. Седякина программы после ее изготовления, а ρ — ресурс, потраченный на ее предварительное тестирование. Используя временные обозначения для времени использования программы по назначению t и времени ее тестирования τ можно записать:

$$P(t,\tau) = e^{-\int_0^t \lambda(z)dz \times e^{-\int_0^\tau \nu(z)dz}}, \quad (2)$$

где λ — интенсивность возникновения ошибки в программе, а ν — интенсивность обнаружения и устранения дефекта в программе при ее тестировании до применения.

Если ранее вероятность безошибочной работы программы мы обозначали, как $P(t)$ и называли ее безотказностью, то в условиях получения формулы (2) будем вероятность $P(t,\tau)$ называть вероятностью работоспособности программы. С ней свяжем понятие ресурса работоспособности программы, который, в отличие от ресурса надежности, будем понимать так:

$$r(t,\tau) = \int_0^t \lambda(z)dz \times e^{\int_0^\tau \nu(z)dz}.$$

В приведенных выражениях временные показатели надежности (работоспособности) следует сопровождать показателями условий их получения или, точнее, физического нагружения, как это, например, делается в формуле ресурса надежности введением величины нагружения ε. Мы этого не делаем, чтобы не усложнять математические выражения.

Исследование влияния процесса тестирования программы на ее работоспособность

Программа расчета надежности нами рассматривается как один из возможных объектов расчета надежности в технике. Рассмотрим упражнение в виде выполнения численных расчетов простого примера с применением обобщенной формулы D. Musa с целью ее графического представления. Итак, имеем формальное представление формулы в виде формулы [4]:

$$P(r,\rho) = e^{-r \times e^{-\rho}}, \quad (3)$$

в которой r — ресурс надежности программы в смысле Н. М. Седякина, а ρ — дополнительный потенциальный ресурс работоспособности, вводимый в разработанную программу в процессе ее предварительных доработок или тестирования. Формальное количественное представление ρ выражается так же, как и ресурс r, но имеет совершенно другое смысловое содержание.

В выполняемом примере полагаем, что значения r изменяются непрерывно в некотором диапазоне значений, а значения ρ принимают значения $\rho = 0$, $\rho 1 = 1$, $\rho 2 = 5$. При этих значениях ρ вероятности успешной работоспособности соответственно принимают следующие выражения:

$$P0(r,\rho) = e^{-r \times e^{-\rho}},$$
$$P1(r,\rho 1) = e^{-r \times e^{-\rho 1}}, \quad (4)$$
$$P2(r,\rho 2) = e^{-r \times e^{-\rho 2}}.$$

На рисунке 1 представлены графики вероятностей (4).

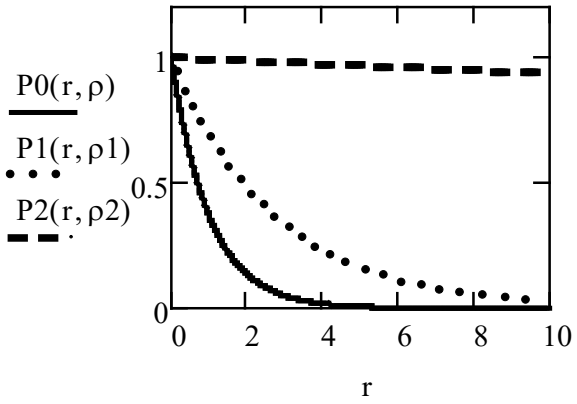


Рис. 1. Вероятности успешной работоспособности программы

Из данного рисунка следует, что вероятность сохранения успешной работоспособности программы в зависимости от величины предварительно введенного потенциального ресурса ρ до начала ее применения возрастает с его увеличением. Далее рассмотрим зависимость полного ресурса работоспособности программы от ее ресурса надежности Н. М. Седякина при различных значениях введенного потенциального ресурса. Полный ресурс работоспособности программы представляется следующими выражениями:

$$R(r,\rho) = r \times e^{-\rho},$$
$$R(t,\tau) = \left(\int_0^t \lambda(z)dz\right) \times e^{-\int_0^\tau \mu(u)du}. \quad (5)$$

Рассмотрим три значения величины полного ресурса работоспособности программы при указанных выше значениях потенциального ресурса $\rho = 0$, $\rho 1 = 1$, $\rho 2 = 5$ по первой формуле из выражений (5). Они представляются такими выражениями:

$$R0(r) = r \times e^{-\rho},$$
$$R1(r) = r \times e^{-\rho 1}, \quad (6)$$
$$R2(r) = r \times e^{-\rho 2}.$$

На рисунке 2 представлены графики зависимостей (6). Они позволяют сделать вывод о том, что в зависимости от значения величины ресурса надежности Н. М. Седякина

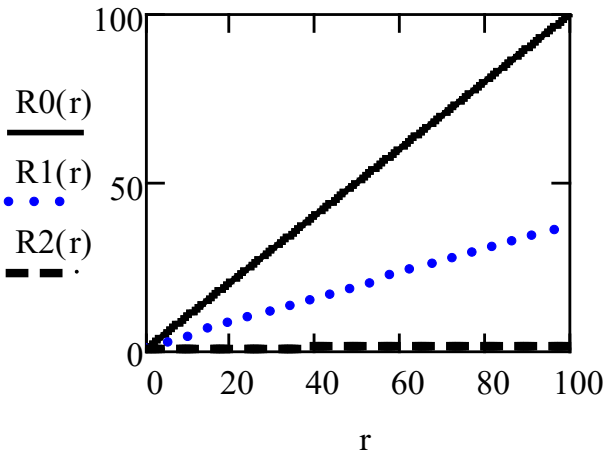


Рис. 2. Графики зависимостей полного ресурса работоспособности от величины ресурса надежности

программы значения полного ресурса ее работоспособности увеличиваются. Это определяется тем, что величина выработанного программой ресурса увеличивается. При этом величина выработанного полного ресурса работоспособности программы наблюдается тем меньше, чем большее значение у нее предварительно введенного потенциального ресурса. Иначе можно утверждать, что чем больше величина предварительно введенного в нее потенциального ресурса до применения программы по назначению, тем больше способность программы сохранять меньшую величину вырабатываемого ею ресурса при эксплуатации.

Для простоты восприятия эффекта рассмотрим простой расчетный пример. Пусть величина ресурса надежности изготовленной программы (ресурса Н. М. Седякина) без ее профилактического обслуживания (тестирования) равна $r = 0{,}5$. Какое представление мы получим, если будем вводить в нее потенциальный ресурс величиной ρ? На рисунке 3 показана зависимость вероятности успешной реализации программы в данном случае, которая вычислена по формуле:

$$P(r,\rho) = e^{-r \times e^{-\rho}}, \quad r = 0{,}5, \quad \rho = 1,\ 0{,}1, \ldots, 10\,.$$

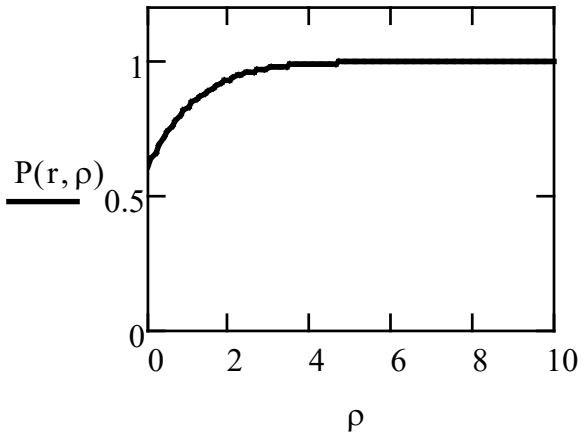


Рис. 3. Вероятность успешной реализации программы

На рисунке 4 показан выигрыш по вероятности отказа реализации программы в зависимости от величины введенного потенциального ресурса в ее структуру. Данный показатель широко использовался профессором А. М. Половко при сравнении качества резервированных структур для повышения их надежности по сравнению с не избыточными, не резервированными структурами. Величина вероятности успешной реализации программы стремится к единичному значению с увеличением введенного в нее потенциального ресурса.

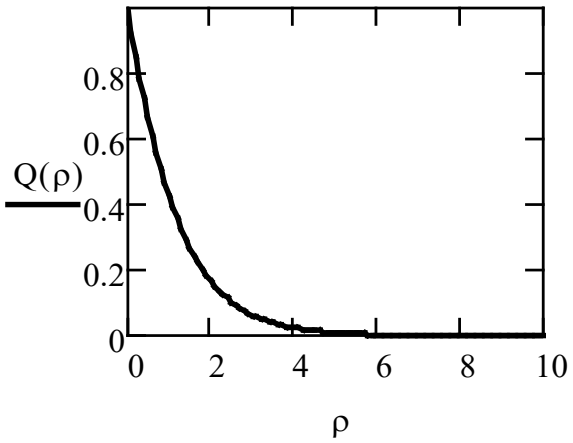


Рис. 4. Вероятность отказа

Из рисунка 4 следует, что вероятность неуспешной реализации программы с введенным потенциальным ресурсом по сравнению с программой без ресурса стремится к нулевому значению тем интенсивнее, чем более значительна величина введенного ресурса.

Обобщение понятия ресурса надежности Н. М. Седякина на основе вероятностной модели D. Musa до понятия ресурса работоспособности живого организма и его значение

Подведем условный вывод о кратком поэтапном развитии теории надежности, представленных в виде двух последовательных этапов, соответствующих появлению двух количественных характеристик или показателей Н. М. Седякина и D. Musa. Надежность — это свойство объекта сохранять его работоспособность в определенных условиях в течение определенного времени. Имевшееся ранее различие к подходу теории надежности в формах математического и физического считаем несущественным, так как они преследовали одну и ту же цель и только дополняли и обогащали друг друга.

На первом этапе развития теории надежности профессор Н. М. Седякин предложил обобщенный интегральный показатель надежности технических объектов, названный им ресурсом надежности на основе существования реальных физических и вероятностных (статистических) данных.

На втором этапе развития теории надежности, связанным с такими процессами, как доработка объектов по выявлению и устранению дефектов техники до ее применения по назначению, а также тестирование сложных программных средств, зарубежный ученый D. Musa предложил новую математическую модель, включающую количественный учет дополнительного экспериментального исследования объектов с целью повышения их работоспособности в процессе применения по назначению.

Это рассмотрение двух совместных этапов количественного оценивания надежности технических и программных объектов позволило ввести обобщенный показатель работоспособности в виде выражения (3). Этот показатель может быть применен не только для технических и программных средств техники, но и для информационных средств, какими являются и живые организмы, а именно человек и человеко-машинные комплексы. Действительно, природа создает человека, наделяет его в определенных условиях жизненного существования конечным статистическим (вероятностным) ресурсом. Это право дано природе в виде первого этапа для успешного существования. Его можно определить количественно в смысле ресурса надежности Н. М. Седякина. Вступая в жизнь с этим ресурсом, человек вступает в реализацию второго этапа жизни. Его работоспособность при данном ресурсе первого этапа на втором этапе будет существенно зависеть от личного поведения. Этим поведением он может управлять, выбирая и создавая свой комплекс условий существования. Человек сам может создавать себе программу существования, выбирать по своему усмотрению те жизненные факторы, которые помогут ему способствовать достижению цели. Это во многом определяется и теми условиями жизни, в которые он попадает.

Как можно трактовать рассматриваемый обобщенный показатель работоспособности живого организма, человека? Система природы с ее условиями порождает человека с определенным ресурсом надежности его деятельности и жизни. Этот ресурс количественно статистически конечен из-за воздействия внешних и внутренних факторов. Выбирая жизненную стратегию переходом ко второму этапу жизни, он может управлять величиной потенциально вводимого ресурса работоспособности, вводя «доработки» и «процесс тестирования и устранения» этим некоторых определенных нежелательных событий жизни. В формуле (3) в показателе степени у ресурса надежности Н. М. Седякина введен сомножитель $e^{-\rho}$, означающий уменьшение утраченного ресурса надежности Н. М. Седякина тем больше, чем больше величина введенного предварительно потенциального ресурса. Физически это означает прореживание существующих вредоносных событий по мере введения значения и величины потенциального ресурса. Величина этого потенциального ресурса уже не может содержать дополнительного экспоненциального сомножителя, который мог бы только реально ухудшить ситуацию с наступлением новых во времени жизни человека событий. Величина этого потенциального ресурса может быть представлена только суммой частных составляющих потенциального ресурса, вводимых человеком в процессе его жизни. Формально это представимо так:

$$\rho = \sum_{i=1}^{\hat{n}} \rho_i,$$

где $\hat{n}$ — случайное число событий, а ρ_i — величина частного жизненного ресурса. Если бы мы ввели величину для сомножителя показателя ресурса ρ в виде $e^{+\gamma}$, то пришли бы к нереальному выводу о единичной вероятности существования живого организма. Если ρ представить в виде суммы частных ресурсов, то это может позволить решать новые реальные для жизни человека задачи обеспечения его работоспособности с учетом различных затрат и факторов жизни. Но это представляется нам как задача будущего исследования.

Заключение

Выделено двухэтапное событие, имевшее место в процессе развития теории надежности систем. Первый этап связан с введением Н. М. Седякиным понятия ресурса надежности объекта как основного показателя надежности систем, который объект имеет и вырабатывает во время применения по назначению.

Второй этап связан с явлением увеличения продолжительности действия объекта за счет устранения его возможных отказов и дефектов благодаря доработкам технического объекта и тестированию функционирования программного обеспечения объекта перед применением по назначению.

На основе выделения этих двух этапов предложено количественное обобщение их частных показателей в один общий показатель — ресурс работоспособности объекта. Выполнено исследование свойств этого показателя и предложены рекомендации по его использованию и дальнейшему развитию в стратегиях жизненной деятельности человека с целью повышения его работоспособности.

Предложенная модель оценивания и обеспечения работоспособности может быть использована при разработке государственной комплексной математической модели противодействия пандемии, которая в настоящее время находится в стадии создания.

Литература

1. Седякин, Н. М. Об одном физическом принципе теории надежности // Известия Академии наук СССР. Техническая кибернетика. 1966. № 3. С. 80–87.

2. Смагин, В. А. Обобщение физического принципа теории надежности профессора Н. М. Седякина // Информация и космос. 2006. № 3. С. 71–78.

3. Смагин, В. А. Модели оценивания надежности элементов на основе форсирования испытаний // Смагин, В. А. Техническая синергетика. Выпуск 1. Вероятностные модели сложных систем: Монография. — Санкт-Петербург: ВКА им. А. Ф. Можайского, 2004. — С. 23–29.

4. Musa, J. D. A Theory of Software Reliability and Its Application // IEEE Transactions on Software Engineering. 1975. Vol. SE-1, Is. 3. Pp. 312–327.
DOI: 10.1109/TSE.1975.6312856.

DOI: 10.24412/2413-2527-2022-230-5-9

Generalization of N. M. Sedyakin's Concept of Reliability Resource of a Technical System to the Working Capacity Resource of a Living Organism

Honoured Science Worker Grand PhD V. A. Smagin, G. L. Yakubovich
Mozhaisky Military Space Academy
Saint Petersburg, Russia
va_smagin@mail.ru, g.l.yakubovich@gmail.com

Abstract. **A quantitative generalization of the concept of professor N. M. Sedyakin about the reliability resource of an object to the resource of human performance is proposed. The first stage is connected with maintaining the concept of an object reliability resource as the main indicator of systems reliability, which an object has and develops during its intended use. The second stage is associated with the phenomenon of increasing the duration of the object by eliminating its possible failures and defects due to the refinement of the technical object and testing the operation of the object's software before using it for its intended purpose. Based on the selection of two stages, a quantitative generalization of their particular indicators into one general indicator is proposed — the resource of the object's operability. A study of the properties of this indicator has been carried out and recommendations have been proposed for its use and further development in the strategies of human life activity in order to increase its efficiency.**

Keywords: **resource of reliability, likelihood model, resource of working capacity of the person, vital strategy, filled resource.**

REFERENCES

1. Sedyakin N. M. About One Physical Principle of Reliability Theory [Ob odnom fizicheskom printsipe teorii nadezhnosti], *Proceedings of the Academy of Sciences of the USSR. Technical Cybernetics [Izvestiya Akademii nauk SSSR. Tekhnicheskaya kibernetika]*, 1966, No. 3, Pp. 80–87.

2. Smagin V. A. Generalization of the Physical Principle of the Reliability Theory by Professor N. M. Sedyakin [Obobshchenie fizicheskogo printsipa teorii nadezhnosti professora N. M. Sedyakina], *Information and Space [Informatsiya i kosmos]*, 2006. No. 3, Pp. 71–78.

3. Smagin V. A. Models for Estimating the Reliability of Elements Based on Forcing Tests [Modeli otsenivaniya nadezhnosti elementov na osnove forsirovaniya ispytaniy]. *In: Smagin V. A. Technical synergy. Issue 1. Probabilistic models of complex systems: Monograph [Tekhnicheskaya sinergetika. Vypusk 1. Veroyatnostnye mo-deli slozhnykh sistem: Monografiya]*. Saint Petersburg, Mozhaisky Military Space Academy, 2004, Pp. 23–29.

4. Musa J. D. A Theory of Software Reliability and Its Application, *IEEE Transactions on Software Engineering*, 1975, Vol. SE-1, Is. 3, Pp. 312–327.
DOI: 10.1109/TSE.1975.6312856.

DOI: 10.24412/2413-2527-2022-230-10-18

Математическая модель процесса плавления высоковязкого нефтепродукта при выгрузке из котла вагона-цистерны

д.т.н. В. И. Моисеев, В. А. Ксенофонтова
Петербургский университет путей сообщения
Императора Александра I
Санкт-Петербург, Россия
moiseev_v_i@list.ru, koc-vera@yandex.ru

к.т.н. Т. А. Комарова
Научно-внедренческий центр «Дисперсные системы»
Санкт-Петербург, Россия
komarova_tanusha@mail.ru

Аннотация. **В статье предлагается математическая модель процесса решения тепловой задачи, связанной с плавлением высоковязких нефтепродуктов при их сливе и очистке котла вагона-цистерны от их остатков. Застывший нефтепродукт имеет форму выпукло-вогнутого сегмента. Для построения общей модели тепловых процессов внутри цистерны область размыва максимально упрощается с использованием метода конформных отображений. Процесс определения границ фазового перехода разбивается на две подзадачи: внутри области размыва распределение температур описывается стационарными решениями, на границе — нелинейная стационарная задача, решаемая с использованием численных методов.**

Ключевые слова: **математическая модель, распределение температуры, фазовый переход, плавление затвердевшего слоя, метод конформных отображений, метод прямых, вязкие нефтепродукты.**

Введение

Одним из актуальных направлений современной математики и численных методов решения различных прикладных задач является изучение нелинейных математических моделей физических явлений. К ним относятся и разнообразные тепловые задачи о фазовых превращениях. В классической постановке задача о замерзании воды (таяния льда) была решена Йозефом Стефаном (Jožef Štefan) в 1889 году и с тех пор носит его имя [1, 2].

Задача Стефана

Простейший вариант математической модели одномерной задачи Стефана поясняет рисунок 1 [3]. Обозначая индексами «1» и «2» величины, относящиеся к твердой и жидкой фазам, отметим, что коэффициенты теплопроводности λ, Вт/(м×°С), удельной теплоемкости C, Дж/(кг×°С), и плотности ρ, кг/м^3, а также объединяющий их коэффициент температуропроводности a, м^2/с,

$$a = \frac{\lambda}{C\rho},$$

у твердой и жидкой фаз различны, но не меняются с температурой: $\lambda_1 \neq \lambda_2$; $C_1 \neq C_2$; $\rho_1 \neq \rho_2$; $a_1 \neq a_2$.

В начальный момент времени твердая фаза отсутствует, а в жидкой фазе имеется некоторое исходное распределение температур:

$$T_2(x,0) = T_н \quad 0 \le x \le l\,. \qquad (1)$$

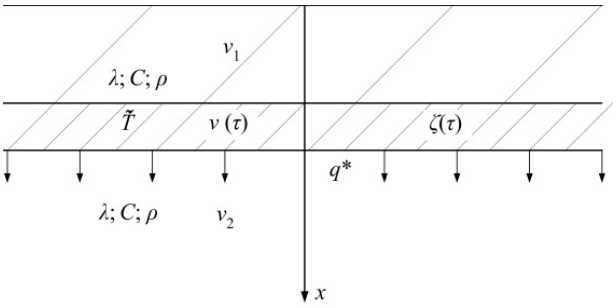


Рис. 1. Схема тепловой задачи Стефана для плавления кристалла

На поверхности ($x = 0$) поддерживается постоянная отрицательная температура, обеспечивающая отток теплоты от фронта кристаллизации через затвердевший слой:

$$T_1(0,\tau) = \Psi_1(\tau); \quad 0 \le x \le l\,. \qquad (2)$$

На фронте кристаллизации жидкая и твердая фазы считаются в идеальном тепловом контакте, их температуры одинаковы и равны температуре фазового перехода, и здесь же выделяется скрытая теплота фазового перехода q^*, которая отводится через обе фазы:

$$\lambda_1 \left(\frac{\partial T_1}{\partial x}\right)_{x=\xi(\tau)} - \lambda_2 \left(\frac{\partial T_2}{\partial x}\right)_{x=\xi(\tau)} = q^* \rho \frac{d\xi}{d\tau}; \qquad (3)$$

$$T_2(\xi(\tau),\tau) = T_1(\xi(\tau),\tau) = T_{кр}\,. \qquad (4)$$

Вдали от фронта кристаллизации жидкая фаза сохраняет свою исходную температуру:

$$T_2(l,\tau) = \Psi_2(\tau). \qquad (5)$$

Плотность теплового потока q^*, Вт/(м^2×°С), задается постоянной и описывается законом теплопроводности Фурье:

$$q = -\lambda \times \operatorname{grad} T. \qquad (6)$$

Распределение температур можно найти, решая систему из двух уравнений теплопроводности в частных производных:

$$\frac{\partial T_1}{\partial \tau} = a_1 \frac{\partial^2 T_1}{\partial x^2}; \qquad 0 < x < \xi(\tau);$$

$$\frac{\partial T_2}{\partial \tau} = a_2 \frac{\partial^2 T_2}{\partial x^2}; \qquad \xi(\tau) \le x < l\,. \qquad (7)$$

Система уравнений (2)–(7) позволяет найти распределение температур в твердой и жидкой фазах $T_1(x, \tau)$, $T_2(x, \tau)$, а также закон движения фронта кристаллизации $\xi(\tau)$.

Для одномерного варианта задачи известно решение [4–6], записанное с использованием безразмерных температурных параметров Θ_1 и Θ_2:

$$\Theta_1 = \frac{T_1 - T_C}{T_{КР} - T_C};\ \Theta_2 = \frac{T_2 - T_H}{T_{КР} - T_H} \qquad (8)$$

и функции, называемой «интегралом ошибок» [4–6]:

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt\ ;\ \operatorname{erfc} x = 1 - \operatorname{erf} x.$$

Решение имеет вид:

$$\Theta_1 = \frac{\operatorname{erf}\left(\frac{x}{2\sqrt{a_1 \tau}}\right)}{\operatorname{erf}\left(\frac{\beta}{\sqrt{2a_1}}\right)};\ \Theta_2 = \frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{a_2 \tau}}\right)}{\operatorname{erfc}\left(\frac{\beta}{\sqrt{2a_2}}\right)}. \qquad (9)$$

Здесь β — постоянная величина, определяемая из трансцендентного уравнения, которое получается подстановкой (8) и (9) в равенство (7):

$$\frac{2(T_{кр} - T_C)}{\sqrt{\pi}} \times \left\{ \left[\frac{\lambda_1}{\sqrt{a_1}} \times \frac{\exp(-\beta^2/2a_1)}{\operatorname{erf}(\beta/\sqrt{2a_1})} \right] + \left[\frac{\lambda_2}{\sqrt{a_2}} \times \frac{\exp(-\beta^2/2a_2)}{\operatorname{erfc}(\beta/\sqrt{2a_2})} \right] \right\} = q^* \rho \beta. \qquad (10)$$

Тепловая задача о плавлении затвердевших нефтепродуктов при их выгрузке и очистке котла вагона-цистерны

Модель Стефана описывает плавление кристаллов или затвердевание жидкости. Но она требует пересмотра при описании затвердевания или разжижения вязких нефтепродуктов (ВНП) — мазутов, парафинов, битумов и других коллоидных, бесструктурных систем без четких границ раздела между маловязкой и высоковязкой областями (рис. 2) [7, 8]. Для них вместо термина «фаза» более оправданным является термин «фракция». У нефтепродуктов в отличие от классических двухфазных систем теплопроводность λ и теплоемкость C и плотность ρ не изменяются при плавлении и затвердевании, приходится вводить такой параметр как вязкость ν, которая резко возрастает при охлаждении.

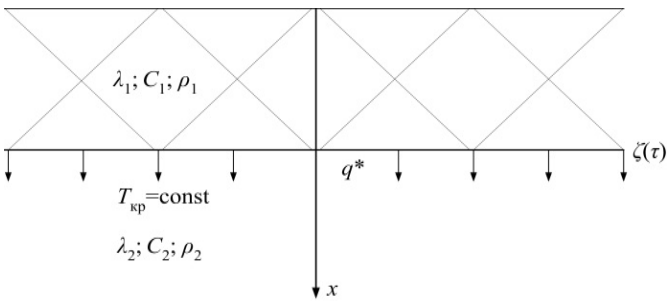


Рис. 2. Схема тепловой задачи по разжижению ВНП

Вместо фронта кристаллизации, определенного координатой x, для ВНП надо ввести понятие «участка затвердевания» конечной длины Δx, вместо температуры фазового перехода $T_{кр}$ — некоторое распределение температур $T_{кр(x)}$ по этому участку. При этом в равенствах (8) и (10) вместо $T_{кр}$ следует ввести усредненную температуру затвердевания $\breve{T}$. Температура $\breve{T}$ зависит и от состава нефтей, из которых был получен нефтепродукт, и от особенностей процесса, в ходе которого он был получен.

Поэтому тепловая задача о плавлении загустевшей фракции ВНП является нелинейной задачей теплопроводности, относящейся к классу Стефана.

Возьмем за направление исследований прикладную задачу выгрузки из вагона-цистерны таких ВНП, как мазуты, крекинг-остатки, парафины и др. Их заливают в котел вагона-цистерны при температуре +100 °С, когда они имеют малую вязкость. При этом во время налива изначально обеспечивается перевод жидкого нефтегруза в стратифицированное состояние.

Плотность ВНП в верхней части котла при этом создается существенно меньшей, чем в нижней его части. Мероприятие блокирует возникновение естественной конвекции горячего ВНП на холодных стенках котла, что резко снижает скорость его охлаждения при перевозках.

На рисунке 3 показано расчетное распределение температур в котельном мазуте М100 по сечению котла вагона-цистерны через 10 суток транспортирования при температуре воздуха T_g = -20 °С. Результаты получены при математическом моделировании тепловых процессов с привлечением пакета программ ANSYS 18.2 [8].

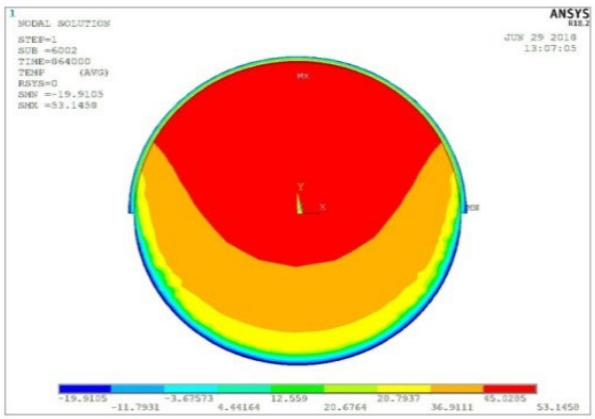


Рис. 3. Распределение температур в массе мазута котельного М100, находящегося в стратифицированном состоянии по сечению котла вагона-цистерны через 10 суток транспортирования при температуре воздуха -20 °С

По нормативам выгрузку мазута М100 проводят самотеком при его температуре +40…45 °С. Из рисунка 2 видно, что за время перевозки свыше 50 % продукта (выделена красным цветом) сохранила высокую температуру и текучесть, достаточные для нормативных условий выгрузки. Около 25…30 % (выделена оранжевым цветом) охладилась до температуры +37 °С, около 10 % (выделено желтым цветом) до температуры +21 °С и находится в вязко-текучем состоянии. Оставшаяся часть 3…5 % (выделена голубым и фиолетовым цветом) охладилась до отри-

цательных температур и, следовательно, затвердела. Модель позволяет видеть и область перехода в высоковязкое состояние толщиной 0,05…0,1 м, она выделена зеленым цветом.

Размыв при выгрузке нужно проводить для массы ВНП, составляющей около 15…20 % всей массы доставленного нефтегруза (выделено фиолетовым, голубым и зеленым цветом). Область размыва в плане чертежа имеет форму выпукло-вогнутого сегмента. Одной его стороной является окружность с радиусом котла R = 1,5 м, а другой — парабола, являющаяся границей затвердевшего слоя мазута. В оптике объекты с такой формой называют «чечевицами», а в теории функций комплексного переменного — «лунками» [9]. Второй термин и будет использоваться ниже.

Решение стационарной задачи о размыве затвердевшего слоя ВНП

Форма и толщина лунок не однозначны, они зависят от вида ВНП, от температуры окружающего воздуха, от продолжительности роста затвердевшего слоя и т. д. Поэтому для построения общей модели тепловых процессов при плавлении высоковязких слоев ВНП необходимо максимально упростить форму области D, где они происходят. Здесь наиболее удобным представляется метод конформных отображений (МКО) [4, 9], применимый для решения стационарных тепловых задач, описываемых уравнением Лапласа:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=0. \qquad (11)$$

Известно, что любая аналитическая функция комплексного переменного удовлетворяет уравнению Лапласа (11) и поэтому в основе МКО лежит сведение заданной сложной области (лунки) к простейшей классической области, для которой решение известно. При этом и уравнение (11) и краевые условия к нему сохраняют свой вид. Поэтому основным преимуществом МКО является возможность получения решений уравнения (11) для практически любой фигуры, заданной на плоскости, в частности для лунки, включающей уже затвердевший, затвердевающий на различной стадии процесс, которые на рисунке 3 выделены бирюзовым, зеленым и желтым цветами.

Решение задачи определения границ областей D фазового перехода внутри котла вагона-цистерны разобьем на две подзадачи. В каждый момент времени область имеет несколько зон различной температуры. Так, одна лунка состоит из внутренней части области

$$D=\{x: u_0<T(x,t)<u_1\}$$

и границ $I_1=\{x: T(x,t)=u_0\}$ и $I_2=\{x: T(x,t)=u_1\}$.

В области D распределение температуры можно описывать стационарными решениями, а на границах I_1 и I_2 — как нелинейную нестационарную задачу, решаемую с использованием численных методов [10, 11].

Для верхней полуплоскости аналитическая функция, описывающая распределение температур и удовлетворяющая краевым условиям, известна [4, 9]. Поэтому и задачу математического описания температурного поля в лунке можно считать решенной.

Введем понятие комплексного потенциала для переноса теплоты. Известно, что плотность теплового потока в среде без тепловых источников определяется уравнением (1). Это уравнение справедливо как для уже застывшего слоя, так и еще не застывающего слоя. Но дивергенция градиента дает оператор Лапласа, а из уравнения теплопроводности (11) следует, что он равен нулю. Тогда

$$\operatorname{div} q=\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}=0, \qquad (12)$$

а так как $\lambda=\text{const}$, то (12) выполняется, когда

$$\frac{\partial q_y}{\partial x}=-\frac{\partial q_x}{\partial y}. \qquad (13)$$

Комплексным потенциалом теплового потока называется функция комплексного переменного $z=x+iy$ следующего вида:

$$\psi(z)=V(x,y)+i\times u(x,y).$$

Ее составляющими являются функции двух переменных, связанные условиями Коши — Римана:

$$\frac{\partial V}{\partial x}=\frac{\partial u}{\partial y}=-\frac{q_y}{\lambda}; \qquad (14)$$

$$\frac{\partial V}{\partial y}=-\frac{\partial u}{\partial x}=-\frac{q_x}{\lambda}. \qquad (15)$$

Дифференцируем обе части равенства (14) по y, а равенства (15) по x. Приравнивая друг другу смешанные производные второго порядка, мы приходим к равенству (13):

$$u(x,y)=Jm|\psi(z)|.$$

Найдем общий вид функции, проводящей отображение лунки $I=\{z: R<|z|<r\}$ (рис. 4) на верхнее полупространство. Граница области I состоит из двух дуг окружностей $I_1=BOA$ и $I_2=BCA$.

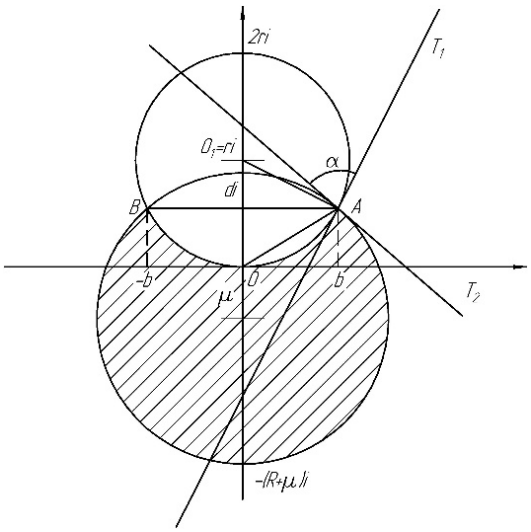


Рис. 4. Выгнуто-вогнутый сегмент, дополнительные построения

Здесь $O_1A=r$ и $O_1O=r$. Тогда треугольник ΔO_1O — равнобедренный. Касательные к окружностям (с радиусом R и r) в точке A обозначим как T_1 и T_2 соответственно, а угол между ними — как α.

Пусть координаты точек A и B есть A (b, d), B (-b, d). Тогда $z_A=b+di, z_B=-b+di$.

Рассмотрим дробно-линейную функцию

$$w_1 = \frac{z-(b+di)}{z+(b-di)}.$$

При этом $w_1(z_A) = 0$, а $w_1(z_B) = \infty$.

Возьмем точку, лежащую на прямой, проходящей через точки A и B, $x+di$. В этом случае

$$w_1(x+di) = \frac{x-b}{x+b} \qquad (16)$$

является вещественным числом.

Так как x может иметь любое числовое значение, то прямая, проходящая через точки A и B, переходит в вещественную ось.

Найдем отображение контуров I_1 = BOA и I_2 = BCA. В силу конформности дробно-линейного отображения контуры I_1 и I_2 переходят в лучи L_1 и L_2 соответственно, исходящие из начала координат. Угол между этими лучами будет равен α.

Возьмем точку $O \in I_1$ и вычислим значение комплексной функции $w_1(O)$:

$$w_1(O) = -\frac{b+di}{b-di} = -\frac{b^2-d^2}{b^2+d^2} - \frac{2bd}{b^2+d^2}i = z_1. \qquad (17)$$

Мнимая и действительная части этой функции равны

$$\mathrm{Im}(z_1) = -\frac{2bd}{b^2+d^2}, \quad \mathrm{Re}(z_1) = \frac{d^2-b^2}{d^2+b^2}. \qquad (18)$$

Обозначим через β угол между лучом L_1 и вещественной осью. Рассмотрим два случая:

1. $d^2 - b^2 < 0 \Rightarrow$ луч лежит в третьем квадранте при этом угол между L_1 и вещественной осью равен

$$\arg\frac{2bd}{d^2-b^2} + \pi;$$

2. $d^2 - b^2 > 0 \Rightarrow$ луч лежит в четвертом квадранте при этом угол между L_1 и вещественной осью равен

$$\arg\frac{2bd}{d^2-b^2} + 2\pi.$$

Отсюда следует, что лунка I отображается функцией w_1 на угол $M = \{w_1 : \beta < \arg w_1 < \beta + \alpha\}$.

Преобразование $w_2 = w_1 e^{-\beta i}$ переводит угол L_2OL_1 на угол $M_1 = \{w_2 : 0 < \arg w_2 < \alpha\}$.

При этом отображение $w_3 = {w_2}^{\pi/\alpha}$ согласно свойству степенной функции комплексной переменной [4, 9] отображает угол M_1 на верхнее полупространство. Окончательное выражение для функции, отображающей лунку на полупространство:

$$w = \left(\frac{z-(b+di)}{z+(b-di)} e^{-\beta i}\right)^{\frac{\pi}{\alpha}}. \qquad (19)$$

Из формулы (19) легко видеть, что нижняя граница лунки I_1 = BOA переходит в положительную часть оси Ox, а I_2 = BCA — в отрицательную часть оси Ox.

Подберем аналитическую в верхней полуплоскости функцию

$$\Psi(z) = V(x,y) + iu(x,y),$$

где $z = x + iy$, мнимая часть которой $\mathrm{Im}(\Psi(z)) = u(x,y)$ на интервале $(-\infty, 0)$ принимает значение u_1, а на интервале $(0, +\infty)$ — значение u_0 (начальные условия). Такую функцию естественно искать в виде

$$u = C_1 \arg(-w) + C_2 \text{ и } \arg(w) = \begin{cases} 0, y > 0, \\ \pi, y < 0. \end{cases} \qquad (20)$$

Постоянные C_1 и C_2 находим из граничных условий:

$$\begin{cases} u_0 = \pi C_1 + C_2, \\ u_1 = C_1 \times 0 + C_2. \end{cases}$$

Тогда

$$\begin{cases} C_2 = u_1, \\ C_1 = (u_0 - u_1)/\pi. \end{cases} \qquad (21)$$

Из (20) и (21) получаем:

$$u = \frac{u_0 - u_1}{\pi}\arg(-w) + u_1. \qquad (22)$$

Теперь построим сам комплексный потенциал, это будет аналитическая функция с мнимой частью вида (22). В качестве такой функции можно выбрать

$$\Psi(w) = \frac{u_0 - u_1}{\pi}\ln(-w) + u_1.$$

При этом получаем окончательное выражение для искомой функции:

$$u(w) = Im\left(\frac{u_0 - u_1}{\pi}\ln(-w)\right) + u_1.$$

Конформные преобразования используют при рассмотрении стационарного распределения температур [3], но когда температура и градиент температуры за расчетный интервал времени на границе среды остаются неизменными, то эта граница называется пассивной. Источники теплоты на ней исключаются из рассмотрения и уравнение теплопроводности принимает форму равенства (11). Допущение о пассивности границ размываемого нефтепродукта оправдывается малыми значениями его теплопроводности λ и сравнительно узкой шириной зоны затвердевания Δx, отмеченной на рисунке 2. Только внутри этой зоны и происходит выделение (поглощение) теплоты q^* фазового перехода парафинов, являющихся одной из компонент ВНП.

Нестационарная часть тепловой задачи плавления ВНП

Рассмотрим вторую часть задачи, а именно границы рассматриваемой лунки. В результате преобразования, описываемого формулой (19) границы лунки перешли в ось Ox, что позволяет рассмотреть задачу как одномерную. Одномерная задача нестационарной теплопроводности имеет следующую математическую модель:

$$\frac{\partial T}{\partial \tau} = a\frac{\partial^2 T}{\partial x^2} + f(x,\tau); \qquad (23)$$

$$T|_{\tau=0} = \phi(x); \quad 0 \le x \le l;$$

$$T(0,\tau) = \Psi_1(\tau); \qquad (24)$$

$$T(l,\tau) = \Psi_2(\tau).$$

Вторую производную заменяем конечной разностью по координате x [12]:

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_k} \approx \frac{T(x_{k+1},\tau) - 2T(x_k,\tau) + T(x_{k-1},\tau)}{h^2}. \qquad (25)$$

При этом задается набор прямых

$$x = x_k = kh\,,\ (k = 0,1,2,\ldots,n+1);\ \ h = \frac{l}{n+1}.$$

Уравнение (23) вместе с краевыми условиями (24) преобразуется в систему уравнений:

$$T_k'(\tau) - \frac{a_k(x_k)}{h^2}[T_{k+1}(\tau) - 2T_k(\tau) + T_{k-1}(\tau)] = f(x_k,\tau)\,, \quad k = 1,2,\ldots,n;$$

$$T_0(\tau) = \psi_1(\tau);\ \ T_{n+1}(\tau) = \psi_2(\tau)$$

Конечно-разностное уравнение (25) аппроксимирует вторую производную с точностью до h^2, но эту точность можно поднять, раскладывая функцию $T(x, \tau)$ в окрестностях точки x_k в ряд Тейлора. После аналогичных преобразований конечно-разностный аналог уравнения (23) приводится к виду [12]:

$$\frac{5}{6}T_k'(\tau) + \frac{1}{12}\left[T_{k+1}'(\tau) + T_{k-1}'(\tau)\right] - \frac{a_k(x_k)}{h^2}[T_{k+1}(\tau) - 2T_k(\tau) + T_{k-1}(\tau)] =$$

$$= \frac{5}{6}f_k(\tau) + \frac{1}{12}[f_{k-1}(\tau) + f_{k+1}(\tau)]\,, \quad k = 1,2,\ldots n.$$

В последнем варианте точность аппроксимации составляет уже h^4.

Более общую математическую модель нестационарных одномерных процессов с фазовыми переходами выражают уравнением теплопроводности, записанным так:

$$\frac{\partial T}{\partial \tau} = a\left[\frac{\partial^2 T}{\partial x^2} + \frac{\upsilon}{x}\frac{\partial T}{\partial x}\right] + f(x,\tau)\,,\ \ z < x < l \qquad (26)$$

Сюда же включают начальное условие (27) и три краевых условия (16)–(18):

$$T|_{\tau=0} = \phi(x)\,, \quad 0 \le x \le l\,; \qquad (27)$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=l} = f(T,\tau)\,, \quad \tau > 0\,; \qquad (28)$$

$$T|_{x=\varsigma} = T_{\text{пл}} = \text{const}\,, \quad \tau > 0. \qquad (29)$$

Скорость движения участка затвердевания Δx будет некоторой функцией температуры, координаты и времени:

$$\frac{dx}{d\tau} = \chi\left(T, \frac{\partial T}{\partial x}, \varsigma, \tau\right) \ \text{ при } \ x = \varsigma\,, \quad \tau > 0. \qquad (30)$$

В уравнении теплопроводности (26) число υ имеет значения υ =0, υ =1 и υ=2 в зависимости от того, в какой системе координат (прямоугольной, цилиндрической или сферической) оно рассматривается. Функции φ, χ и f могут быть произвольными, так как метод прямых не накладывает ограничений на характер этих зависимостей. При рассмотрении системы уравнений (26)–(30) применяют подстановку Ландау:

$$y = \frac{l - x}{l - \varsigma}. \qquad (31)$$

$$y|_{x=l} = 0;\ \ y|_{x=z} = 1. \qquad (32)$$

После перехода к новой переменной (37) получаем равенства:

$$\frac{\partial T}{\partial \tau} = p\frac{\partial^2 T}{\partial y^2} - q\frac{\partial T}{\partial y}\,, \quad 0 < y < 1\,, \quad \tau > 0\,;$$

$$p = \frac{a}{(l-\varsigma)^2};$$

$$q = \frac{a\varsigma}{(l-\varsigma)\times[l - y(l-\varsigma)]} + \frac{y\frac{d\varsigma}{d\tau}}{l-\varsigma}\,; \qquad (33)$$

$$\frac{\partial T}{\partial y} = -(l-\varsigma)f(T,\tau)\,, \quad y = 0, \quad \tau > 0. \qquad (34)$$

Источники теплоты q^* в ВНП, как отмечалось выше, проявит себя только тогда, когда к данной точке среды подойдет участок затвердевания Δx; вне этого участка объемная плотность тепловыделения равна нулю:

$$T = T_{\text{пл}}(y)\,, \ \ 0 \le y \le 1\,, \ \ \tau = 0.$$

В соответствии с методом прямых плоскость (y, τ) разбивается на (m +1) участков прямыми линиями:

$$y = j\Delta y\,, \quad j = 1,2,\ldots m.$$

Частные производные первого и второго порядка $\partial T/\partial y$ и $\partial^2 T/\partial y^2$ заменяются разностями

$$\frac{\partial T}{\partial y} \approx \frac{T(y_{i+1}) - T(y_i)}{2\Delta y}\,; \qquad (35)$$

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_{y=y_i} \approx \frac{T(y_{i+1}) - 2T(y_i) + T(y_{y-1})}{\Delta y^2}\,, \qquad (36)$$

а производная по времени $\partial T/\partial\tau$ заменяется той же производной, рассчитываемой вдоль i-й прямой $\partial T_i/\partial\tau$. После подстановки (35) и (36) в (32) получаем конечно-разностный аналог уравнения теплопроводности (24), построенный в соответствии с методом прямых:

$$\frac{\partial T_i}{\partial \tau} = p\left(\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta y^2}\right) - q\left(\frac{T_{i+1} - T_{i-1}}{2\Delta y}\right),\ i = 1,2,\ldots m-1. \qquad (37)$$

Здесь p и q по равенству (25) для соответствующих значений y и z [12]. Разностное уравнение (37) считается справедливым на прямой i = 0. Температура T_{-1} относится уже к области внешней по отношению к рассматриваемой ее находят подстановкой (35) в граничное условие (33):

$$T_{-1} = T_1 - 2\Delta y(l - z)f(T_0,\tau). \qquad (38)$$

Итак, задача свелась к нахождению решения системы (m+1) уравнений, из которых (m-1) образует подсистему (37), еще одно уравнение (38) следует из этой же системы для i = 0, и последнее получается из (34) после разрешения его относительно ζ.

Функция, описывающая распределение температур в лунке, аналогична функции (9):

$$\operatorname{erf}\left(\frac{x}{2\sqrt{a\tau}}\right)=\frac{\mathrm{T}(\mathrm{x},\tau)-\mathrm{T}_{\mathrm{ст}}}{\mathrm{T}_0-\mathrm{T}_{\mathrm{ст}}}.$$

В комплексной форме ее можно записать следующим образом:

$$w(x)=e^{-x^2}\operatorname{erfc}\left(\frac{-ix}{2\sqrt{a\tau}}\right)=\frac{2e^{-x^2}}{\sqrt{\pi}}\int_{-\frac{ix}{2\sqrt{a\tau}}}^{+\infty}e^{-t^2}dt=$$

$$=\frac{\mathrm{T}(x,\tau)-\mathrm{T}_{\mathrm{ст}}}{\mathrm{T}_0-\mathrm{T}_{\mathrm{ст}}}.$$

Отметим, что z из (17) не выражается в явном виде через w (19), и при расчетах для заданного z надо найти значение w по формуле (20), чтобы получить соответствующее значение потенциала переноса в рассматриваемой плоскости.

На основе этого алгоритма была построена программа для ЭВМ «STEFAN» написанная на языке FORTRAN стандарт 77 (оболочка FPS-1.0) [13–15]. Результаты выполненных расчетов показаны на рисунках 5–7.

На рисунке 5 внутренняя поверхность затвердевшего слоя обозначена позицией 1, наружная поверхность слоя — позицией 21.

При выполнении расчетов принято, что:

- температура мазута внутри цистерны (кривая 1) — +25 ºС;
- температура окружающей среды и, следовательно, слоя мазута на стенках котла (кривая 21) равна минус 40 ºС;
- начальная толщина застывшего слоя мазута — 100 мм;
- коэффициент теплоотдачи с внутренней стороны слоя $\alpha_{вн}$ = 10 Вт/(м^2×К);
- теплоотдача с наружной стороны $\alpha_{нар}$=15 Вт/(м^2×К).

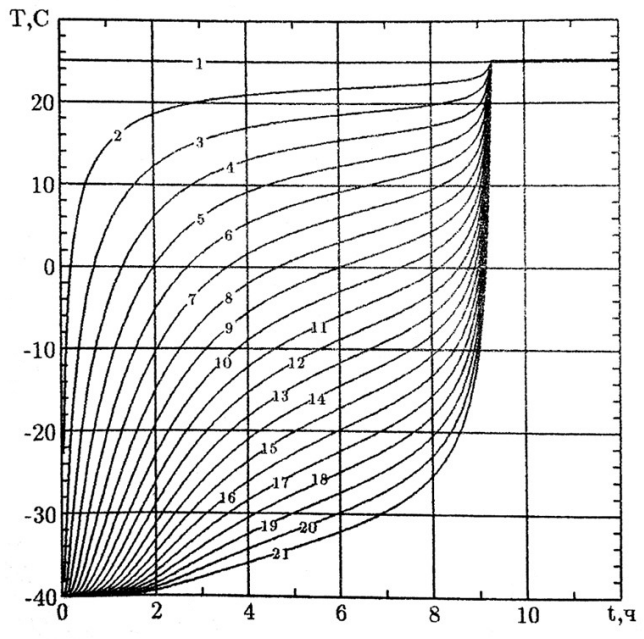


Рис. 5. Изменение распределения температур в затвердевшем слое мазута при его разогреве и плавлении с учетом фазового перехода (расчет кривых отвечает шагу 1/20 толщины слоя)

Кривая 2 определяет разогрев внутреннего слоя мазута толщиной около 5 мм. Она показывает, что температура плавления мазута, равная +15ºС достигается приблизительно через один час. Прогрев и плавление более глубоких слоев (кривые 3 и 4) завершится через 4 и 6 часов соответственно. К этому времени толщина затвердевшего слоя окажется приблизительно равной 50…60 мм. Дальше процесс разогрева и плавления резко ускоряется за счет уменьшения термического сопротивления затвердевшего слоя. Затвердевший слой будет полностью размыт через 9 часов и это при очень низкой температуре атмосферного воздуха.

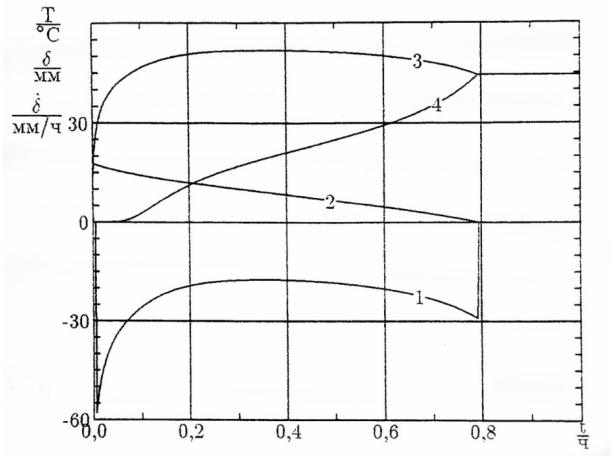


Рис. 6. Изменение во времени толщины застывшего разжижающегося слоя мазута и температуры его внутренней поверхности в зависимости от времени с учетом фазовых превращений

При выполнении расчетов принято, что:

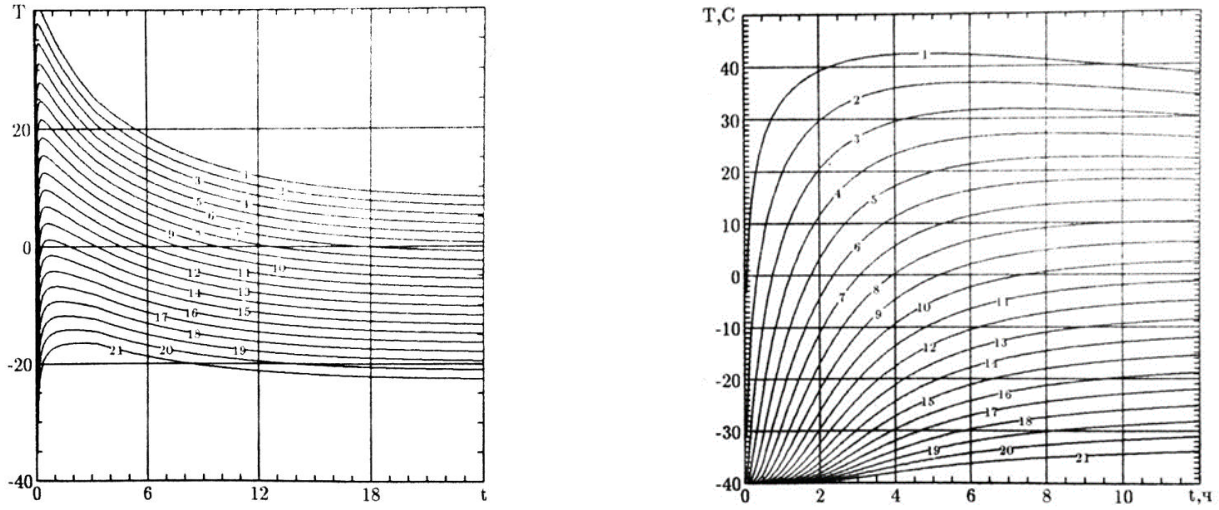
- температура окружающей среды и, следовательно, слоя мазута на стенках котла равна 0ºС;
- температура внутри котла вагона-цистерны — +63 ºС;
- начальная толщина застывшего слоя мазута — 17 мм;
- теплопроводность слоя — 0,1 Вт/(м×К);
- коэффициент теплоотдачи с внутренней стороны слоя $\alpha_{вн}$ =36 Вт/(м^2×К);
- теплоотдача с наружной стороны $\alpha_{нар}$=15 Вт/(м^2×К).

На рисунке 6 кривая 1 определяет скорость толщины застывшего разжижающегося слоя мазута. Она показывает, что в первые часы (моменты времени) процесс проходит достаточно активно (кривая резко идет вверх), далее, по мере уменьшения затвердевшего слоя (кривая 2) скорость становится меньше. Процесс разжижения прекращается через 0,8 часа, когда кривая 2 приближается к нулю. При этом внутренняя и внешняя температуры (кривые 3 и 4) уменьшаются и увеличиваются соответственно. И принимают через 0,8 часа значение 45ºС, достаточное для выгрузки мазута самотеком.

На рисунке 7 внутренняя поверхность затвердевшего слоя обозначена кривой 1, наружная поверхность слоя — кривой 21.

При выполнении расчетов принято, что:

- температура мазута внутри цистерны (кривая 1) — +65 °С;
- температура окружающей среды и, следовательно, слоя мазута на стенках котла (кривая 21) равна минус 40 °С;



а) б)

Рис. 7. Распределение температур через 1/20 толщины пленки
а — для застывшего слоя толщиной 10 мм; *б* — для застывшего слоя толщиной 100 мм

• коэффициент внутренней теплоотдачи $\alpha_{вн} = 50$ Вт/(м^2×К) для рисунка 7, *а* и $\alpha_{вн} = 10$ Вт/(м^2×К) для рисунка 7, *б*;

• коэффициент внешней теплоотдачи $\alpha_{нар} = 50$ Вт/(м^2×К) для рисунка 7, *а* и $\alpha_{нар} = 15$ Вт/(м^2×К) для рисунка 7, *б*.

Из рисунка 7, *а* видно, что кривая 21, показывающая изменение температуры наружного слоя застывшего нефтепродукта, меняется от минус 40 ºС до приблизительно минус 21 ºС через 12 часов, в то время как кривая 1, показывающая изменение температуры внутреннего слоя застывшего нефтепродукта, меняется от минус 40 °С до приблизительно плюс 10 °С через такое же время. В то же время из рисунка 6, *б* видно, что кривая 21, показывающая изменение температуры наружного слоя застывшего нефтепродукта, меняется от минус 40 ºС до приблизительно минус 35 °С, в то время как кривая 1, показывающая изменение температуры внутреннего слоя застывшего нефтепродукта, меняется от минус 40 °С до приблизительно плюс 41 °С за те же 12 часов. Разница в температуре внутренних и внешних слоев объясняется разницей в толщине затвердевшего слоя и горячим нефтепродуктом в центре котла вагона-цистерны.

Суммарные потери теплоты для процесса, отмеченного на рисунке 7, *а*, составляют $4{,}7\times10^9$ Вт, а для процесса, отмеченного на рисунке 7, *б*, — $1{,}55\times10^9$ Вт.

Такие значения времени разогрева и размыва приблизительно соответствуют практике эксплуатации сливных площадок.

Заключение

Построена нелинейная математическая модель процесса плавления вязких нефтепродуктов при их выгрузке с разогревом из котла вагона-цистерны.

Построенная модель позволяет решать технологические задачи разогрева до разжижения вязких нефтепродуктов при их сливе и очистке котла вагона-цистерны от их высоковязких остатков, а также оценить потребные затраты времени и тепловой энергии.

Литература

1. Данилюк, И. И. О задаче Стефана // Успехи математических наук. 1985. Т. 40, Вып. 5 (245). С. 133–185.

2. Мейрманов, А. М. Задача Стефана / А. М. Мейрманов; АН СССР, Сиб. отд-ние, Ин-т гидродинамики. — Новосибирск: Наука. Сиб. отд-ние, 1986. — 239 с.

3. Назинцев, Ю. Л. Теплофизические свойства морского льда / Ю. Л. Назинцев, Ж. А. Дмитраш, В. И. Моисеев. — Ленинград [Санкт-Петербург]: Изд-во Лен. Гос. ун-та, 1988. — 259 с.

4. Лыков, А. В. Тепломассообмен: Справочник. — 2-е изд., перераб. и доп. — Москва: Энергия, 1978. — 480 с.

5. Лыков, А. В. Теория теплопроводности. — Москва: Высшая школа, 1967. — 600 с.

6. Гребер, Г. Основы учения о теплообмене = Die Grundgesetze der Wärmeübertragung. Dritte völlig neubearbeitete auflage / Г. Гребер, С. Эрк, У. Григулль; пер. с нем. под общ. ред. А. А. Гухмана. — Москва: Издательство иностранной литературы, 1958. — 566 с.

7. Кладов, А. В. Вязкостно-температурные свойства нефтепродуктов / А. В. Кладов, В. П. Коваленко, В. В. Шлячков // Сборник рефератов депонированных рукописей. Серия Б, инв. № Б5591, Выпуск № 69. — Москва: ЦВНИ МО РФ, 2004.

8. Моисеев, В. И. Математические модели и численные методы тепло- и массопереноса при естественной конвекции горячих жидких нефтепродуктов в котле вагона-цистерны / В. И. Моисеев, В. А. Ксенофонтова, Т. А. Комарова // Интеллектуальные технологии на транспорте. 2022. № 1 (29). С. 5–15. DOI: 10.24412/2413-2527-2022-129-5-15.

9. Фукс, Б. А. Функции комплексного переменного и некоторые их приложения / Б. А. Фукс, Б. В. Шабат. — 3-е изд. — Москва: Наука, 1964. — 388 с. — (Физико-математическая библиотека инженера).

10. Пехович, А. И. Расчеты теплового режима твердых тел / А. И. Пехович, В. М. Жидких. — 2-е изд., перераб. и доп. — Ленинград [Санкт-Петербург]: Энергия. Ленингр. отд-ние, 1976. — 352 с.

11. Пестов, К. Н. Об одном численном методе решения задач типа Стефана / К. Н. Пестов, О. Н. Любимова, М. В. Останин // Вестник Чувашского государственного педагогического университета им. И. Я. Яковлева. Серия:

Механика предельного состояния. 2019. № 3 (41). С. 51–60. DOI: 10.26293/chgpu.2019.41.3.002.

12. Кузнецов, Г. В. Разностные методы решения задач теплопроводности: Учебное пособие / Г. В. Кузнецов, М. А. Шеремет. — Томск: Изд-во Томского политехн. ун-та, 2007. — 172 с.

13. Свидетельство о государственной регистрации программы для ЭВМ № 2012619613 Российская Федерация. Программа для расчета тепловых процессов при разогреве и размыве вязких нефтепродуктов, сливаемых из железнодорожной цистерны «STEFAN 1»: опубл. 13.09.2012 / В. И. Моисеев, Д. В. Елисеев.

14. Свидетельство о государственной регистрации программы для ЭВМ № 2012618250 Российская Федерация. Программа расчета скорости движения границы расплава вязких нефтепродуктов, разогреваемых перед сливом в железнодорожной цистерне: опубл. 12.09.2012 / В. И. Моисеев.

15. Свидетельство о государственной регистрации программы для ЭВМ № 2012616057 Российская Федерация. Программа для расчета распределения температур в темных нефтепродуктах при их выгрузке из ж.-д. цистерны в зимних условиях: опубл. 03.07.2012 / В. И. Моисеев, Д. В. Елисеев.

DOI: 10.24412/2413-2527-2022-230-10-18

Mathematical Model of Melting Process of High Viscosity Oil Product Upon Discharge from of Tank Wagon

Grand PhD V. I. Moiseev, V. A. Ksenofontova
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
moiseev_v_i@list.ru, koc-vera@yandex.ru

PhD T. A. Komarova
Research and Implementation Center «Dispersed Systems»
Saint Petersburg, Russia
komarova_tanusha@mail.ru

Abstract. **The article offers a mathematical model of the thermal problem connected with melting of highly viscous oil products at their discharge and cleaning of the boiler of the tank wagon from their residues. The frozen petroleum product is in the form of a convex-concave segment. In order to construct a general model of thermal processes inside the tank, the blur area is simplified as much as possible using conformal mapping. The phase transition delimitation process is divided into two subtasks: inside the blur area, the temperature distribution is described by stationary solutions, at the border — a nonlinear stationary problem solved using numerical methods.**

Keywords: **mathematical model, temperature distribution, phase transition, melting of the solidified layer, conformal mapping method, method of straight lines, viscous oil products.**

REFERENCES

1. Danilyuk I. I. On the Stefan Problem [O zadache Stefana], *Russian Mathematical Surveys [Uspekhi matematicheskikh nauk]*, 1985, Vol. 40, Is. 5, Pp. 157–223. DOI: 10.1070/RM1985v040n05ABEH003684.

2. Meyrmanov A. M. The Stefan's Problem [Zadacha Stefana]. Novosibirsk, Nauka Publishers, 1986, 239 p.

3. Nazintsev Yu. L., Dmitrash Zh. A., Moiseev V. I. Thermophysical properties of sea ice [Teplofizicheskie svoystva morskogo lda]. Leningrad [Saint Petersburg], Pushkin Leningrad State University, 1988, 259 p.

4. Lykov A. V. Heat and mass transfer: Reference book [Teplomassoobmen: Spravochnik]. Moscow, Energia Publishing House, 1978, 480 p.

5. Lykov A. V. Theory of thermal conductivity [Teoriya teploprovodnosti]. Moscow, Vysshaya Shkola Publishers, 1967, 600 p.

6. Gröber H., Erk S., Grigull U. Die Grundgesetze der Wärmeübertragung. Dritte völlig neubearbeitete auflage [Osnovy ucheniya o teploobmene]. Moscow, Foreign Literature Publishing House, 1958, 566 p.

7. Kladov A. V., Kovalenko V. P., Shlyachkov V. V. Viscosity-Temperature Properties of Petroleum Products [Vyazkostno-temperaturnye svoystva nefteproduktov], *Collection of abstracts of deposited manuscripts [Sbornik referatov deponirovannykh rukopisey], Series B, No. B5591*, Is. 69. Moscow, Center for Military Scientific Information of the Ministry of Defense of the Russian Federation, 2004.

8. Moiseev V. I., Ksenofontova V. A., Komarova T. A. Mathematical Models and Numerical Methods of Heat and Mass Transfer During Natural Convection of Hot Liquid Petroleum Products in Boiler of Tank Wagon [Matematicheskie modeli i chislennye metody teplo- i massoperenosa pri estestvennoy konvektsii goryachikh zhidkikh nefteproduktov v kotle vagona-tsisterny], *Intellectual Technologies on Transport [Intellektualnye tekhnologii na transporte]*, 2022, No. 1 (29), Pp. 5–15. DOI: 10.24412/2413-2527-2022-129-5-15.

9. Fuks B. A., Shabat B. V. Functions of a complex variable and some of their applications [Funktsii kompleksnogo peremennogo i nekotorye ikh prilozheniya]. Moscow, Nauka Publishers, 1964, 388 p.

10. Pekhovich A. I., Zhidkikh V. M. Calculations of the thermal regime of solids [Raschety teplovogo rezhima tverdykh tel]. Leningrad [Saint Petersburg], Energia Publishing House, 1976, 352 p.

11. Pestov K. N., Lyubimova O. N., Ostanin M. V. The Method of Solving Problems Type of Stefan with Phase Transitions of the First Kind [Ob odnom chislennom metode resheniya zadach tipa Stefana], *Bulletin of the Yakovlev Chuvash State Pedagogical University. Series: Mechanics of Limit State [Vestnik Chuvashskogo gosudarstvennogo pedagogicheskogo universiteta imeni I. Ya. Yakovleva. Seriya: Mekhanika predelnogo sostoyaniya]*, 2019, No. 3 (41), Pp. 51–60. DOI: 10.26293/chgpu.2019.41.3.002.

12. Kuznetsov G. V., Sheremet M. A. Difference methods for solving heat conduction problems: Study guide [Raznostnye metody resheniya zadach teploprovodnosti: Uchebnoe posobie]. Tomsk, Tomsk Polytechnic University, 2007, 172 p.

13. Moiseev V. I., Eliseev D. V. A Program for Calculating Thermal Processes During Heating and Washing Out of Viscous Oil Products Drained from the «STEFAN 1» Railway Tank [Programma dlya rascheta teplovykh protsessov pri razogreve i razmyve vyazkikh nefteproduktov, slivaemykh iz zheleznodorozhnoy tsisterny «STEFAN 1»]. Certificate of State registration of a computer program RU No. 2012619613, published at September 13, 2012.

14. Moiseev V. I. A Program for Calculating the Speed of Movement of the Melt Boundary of Viscous Petroleum Products Heated Before Discharge in the Same Road Tank [Programma rascheta skorosti dvizheniya granitsy rasplava vyazkikh nefteproduktov, razogrevaemykh pered slivom v zheleznodorozhnoy tsisterne]. Certificate of State registration of a computer program RU No. 2012618250, published at September 12, 2012.

15. Moiseev V. I., Eliseev D. V. A Program for Calculating the Temperature Distribution in Dark Oil Products During Their Unloading from the Railway Tanks in Winter [Programma dlya rascheta raspredeleniya temperatur v temnykh nefteproduktakh pri ikh vygruzke iz zheleznodorozhnoy tsisterny v zimnikh usloviyakh]. Certificate of State registration of a computer program RU No. 2012616057, published at July 03, 2012.

DOI: 10.24412/2413-2527-2022-230-19-24

Способ фильтрации данных GPS для цикла движения. Исследование топливной экономичности специального автомобиля

О. А. Дурницын
Тюменский индустриальный университет
Тюмень, Россия
durnitsynoa@mail.ru

Аннотация. **Устройства сбора данных Глобальной системы позиционирования, или GPS, зарекомендовали себя как полезные инструменты для сбора реальных данных о движении и статистики. Данные, собранные этими устройствами, предоставляют ценную информацию при изучении параметров движения транспортных средств. Эти данные имеют неоценимое значение для анализа расхода топлива и производительности транспортных средств. Без надежной информации о скорости и времени достоверность полученных показателей для циклов движения, таких как ускорение, мощность и расстояние, становится сомнительной. В этом исследовании рассматриваются некоторые общие источники ошибок, присутствующие в собранных необработанных данных GPS, и представлен подробный процесс фильтрации, предназначенный для устранения этих проблем, с последующим построением графика влияния ошибок на корректность расчета экономии топлива. Проанализированы данные испытаний для тяжелых условий эксплуатации. Представлены графические сравнения необработанных и отфильтрованных циклов, а также проведен статистический анализ для определения влияния предлагаемого процесса фильтрации на необработанные данные. Наконец, статья завершается оценкой общих преимуществ фильтрации данных GPS и представляет опыт расчета топливной экономичности автобетоносмесителя, для которого был построен цикл.**

Ключевые слова: **GPS data, фильтрация сигнала, фильтрация данных, расход топлива, ездовой цикл.**

Введение

Экономичный характер и простота установки, связанные с системами сбора данных GPS, способствовали тому, что регистрация данных бортовой системы глобального позиционирования быстро стала одним из наиболее популярных методов сбора реальной информации о работе транспортного средства [1–3]. Связанные данные о скорости и времени транспортного средства, полученные этими устройствами, представляют особый интерес при выполнении моделирования транспортного средства и анализа цикла движения [4, 5]. Однако из-за чувствительности этих нижестоящих приложений как к качеству, так и к целостности исходных данных GPS, уникальное рабочее поведение и ошибки, присущие регистраторам данных GPS, вызвали необходимость в новом процессе фильтрации. Идеальным методом фильтрации для улучшения качества необработанных данных GPS является тот, который сводит к минимуму последствия ошибок регистрации данных GPS, таких как внезапная потеря сигнала, скачки данных, белый шум сигнала и колебания нулевой скорости при сохранении целостности исходных данных [6]. Это означает удаление или замену любых ошибочных точек данных при одновременном сведении к минимуму общего объема измененных данных. Идеальный метод фильтрации также должен быть достаточно гибким для применения к данным GPS, полученным с различных устройств. Таким образом, любой предлагаемый метод фильтрации должен учитывать различия в качестве данных, доступных через различные GPS-считыватели данных, а также различные назначения транспортных средств или типы, в которых могут быть установлены регистраторы. Представленный подход к фильтрации данных GPS состоит из трех различных фильтров, разработанных и расположенных специально для удовлетворения этих критериев. В попытке учесть распространенные ошибки, связанные с анализом данных о скорости и времени GPS, в среде STAMM 3.0 был разработан общий метод обработки данных GPS, который использует серию линейно прогрессирующих логических фильтров данных [7, 8].

Чтобы оценить эффективность предлагаемого процесса фильтрации данных GPS, исследователи проанализировали более 170 индивидуальных циклов движения, полученных с помощью GPS-устройств, установленных на борту тяжелых транспортных средств.

Циклы движения, проанализированные в рамках этого исследования, содержат связанную информацию о скорости и времени, собранную с помощью встроенных GPS-систем автоматического сбора данных, и содержат информацию о движении либо за один рабочий день транспортного средства, либо за смену. Были рассчитаны статистические данные, касающиеся количества данных, измененных на каждом этапе фильтрации, и проанализированы результаты процесса фильтрации. В дополнение к изучению величины данных, измененных в процессе фильтрации, были выполнены расчеты анализа приводного цикла для оценки влияния фильтрации на динамику базового цикла. Последующим шагом стала оценка топливной экономичности с построением графика.

Логический подход к фильтрации

Предлагаемый процесс фильтрации для анализа данных GPS о скорости и времени для циклов движения состоит из трех логических фильтров, расположенных в порядке возрастания сложности. Они заключаются в следующем:

1. Удаление повторяющихся записей и данных с отрицательными значениями или дифференциальными временными шагами.

2. Замена внешних значений высокой/низкой скорости.

3. Устранение колебаний сигнала нулевой скорости при остановке транспортного средства.

В качестве начального шага в процессе фильтрации необходимо удалить любые точки данных с повторяющимися значениями времени и точки данных, которые имеют отрицательные или нулевые дифференциальные значения времени (т. е. время в точке 2 меньше времени в точке 1, или время в точке 2 равно времени в точке 1). Чтобы удалить эти точки, фильтр сначала вычисляет разностные значения времени для каждой из точек данных в исходном наборе, а затем удаляет все, у которых разностные значения времени меньше или равны нулю.

Этот шаг должен быть выполнен в первую очередь из-за влияния разницы во времени при расчете ускорения на основе информации о скорости. Если существуют повторяющиеся записи времени, то по определению ускорения из базовой кинематики ускорение в повторяющейся точке будет приближаться к бесконечности. Если бы ошибочная информация о времени не была удалена на этом предварительном этапе, мы не смогли бы выполнить точные вычисления на основе дифференциальной/интегральной информации о времени, и будущие этапы фильтрации, основанные на этой информации, были бы невозможны.

На втором этапе процесса фильтрации любые ошибочные точки данных, такие как скачки скоростных данных с одной выборкой и отсев сигнала с отрицательной скоростью, которые присутствуют в наборе данных, удаляются и заменяются интерполированными данными. Для достижения этой цели каждое значение скорости в наборе образцов обрабатывается через фильтр и сравнивается индивидуально с выбранными предельными значениями высокой/низкой скорости. Если обнаружено, что точка данных находится за выбранными пределами, фильтр заменяет исходную точку выборки данных информацией о скорости, полученной из интерполяции кубического сплайна, полученной из соседних данных [8]. В качестве дополнительного шага в процессе интерполяции отфильтрованная скорость транспортного средства ограничена нижней границей до нуля километров в час, чтобы облегчить моделирование транспортного средства.

Применение контролируемых пользователем ограничений обеспечивает гибкость при обработке данных и улучшает качество результатов, когда известны ограничения скорости транспортного средства. На рисунке 1 ограничение высокой скорости в 75 км/ч использовалось в сочетании с фильтром внешней скорости для удаления ошибочного всплеска сигнала высокой скорости в образце данных GPS о транспортном средстве. Ограничение было применено при условии, что скорость, превышающая 75 км в час, может быть отнесена к всплескам сигнала, а не к нормальной работе. В случае неизвестных транспортных средств рекомендуется выбирать более консервативные высокие/низкие эксплуатационные пределы, чтобы избежать удаления точной информации. Использование регулируемых пределов в фильтре внешних скоростей обеспечивает уникальное преимущество при фильтрации большого количества циклов движения, поскольку пределы фильтрации можно регулировать для отдельных типов транспортных средств.

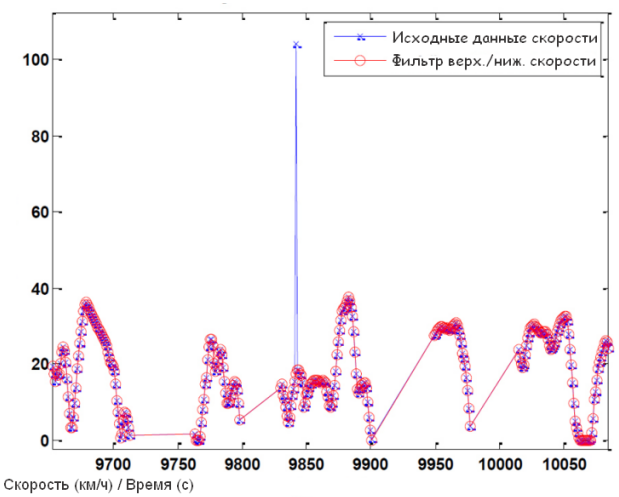


Рис. 1. Результаты фильтрации по удаленной скорости для выборочных данных GPS о транспортном средстве

Устранение колебаний нулевой скорости GPS

На третьем этапе процесса фильтрации устраняется эффект, называемый колебанием нулевой скорости. Колебание нулевой скорости появляется, когда работает GPS-навигатор, а автомобиль останавливается и работает на холостом ходу в течение длительного периода времени. Во время длительных простоев регистраторы данных GPS часто регистрируют очень малое значение скорости (0,1 или 0,2 км/ч) из-за повторного запроса сигнала спутника GPS, который происходит при остановке транспортного средства. Чтобы устранить небольшие колебания скорости транспортного средства, регистрируемые GPS-системы сбора данных в течение этих периодов, фильтр колебаний нулевой скорости проверяет расстояние, пройденное во время каждого отрезка в цикле движения, и сравнивает значение с заданным пользователем пределом. Если расстояние, пройденное в ходе отрезка, ниже предельного значения, вся поездка заменяется значениями данных с нулевой скоростью. Как показано на рисунке 2, применение ограничения расстояния в 0,001 км к выборочным данным GPS приводит к удалению низкоскоростного отрезка. При выполнении традиционных вычислений расстояния на основе интеграции с использованием связанной информации о скорости и времени удаление данных о колебании на нулевой скорости приводит к уменьшению расчетных расстояний, которые в большей степени отражают фактическую эксплуатацию транспортного средства.

Следующий шаг в процессе фильтрации — фильтр ложной нулевой скорости — удаляет одноточечные записи данных с нулевой скоростью, которые являются результатом временного пропадания сигнала GPS. Подобно фильтру удаленной скорости, фильтр с ложным нулем удаляет записи о ложной нулевой скорости, проверяя значение каждой отдельной точки скорости в наборе данных по отношению к ее ближайшим соседям. Если заданное значение записи скорости равно нулю, а соседние точки с каждой стороны отличны от нуля, точка нулевой скорости заменяется точкой, полученной из интерполяции кубического сплайна всего оставшегося набора данных.

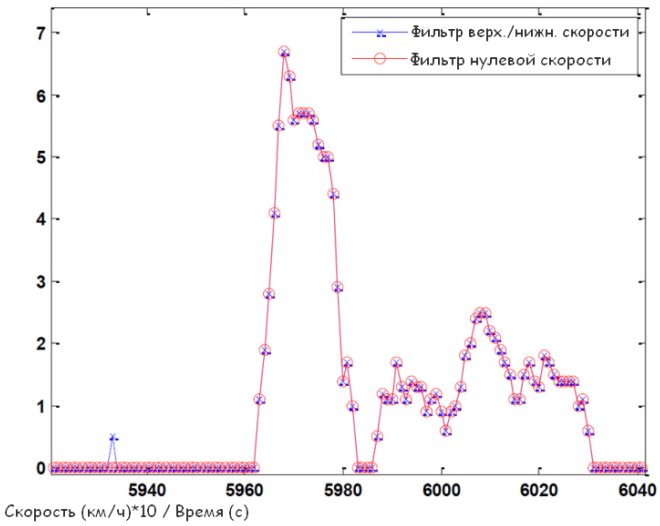


Рис. 2. Результаты фильтра дрейфа колебаний нулевой скорости для выборочных данных GPS транспортного средства

Этот простой фильтр, показанный на рисунке 3, улучшает непрерывность исходных данных, заменяя значительные выбросы в данных данными, более репрезентативными для реальной эксплуатации транспортного средства. Устраняя резкие изменения скорости, улучшаются расчеты энергии при анализе цикла движения; что еще более важно, в сценариях моделирования транспортного средства модель способна точно следовать заданной траектории скорости и времени [9].

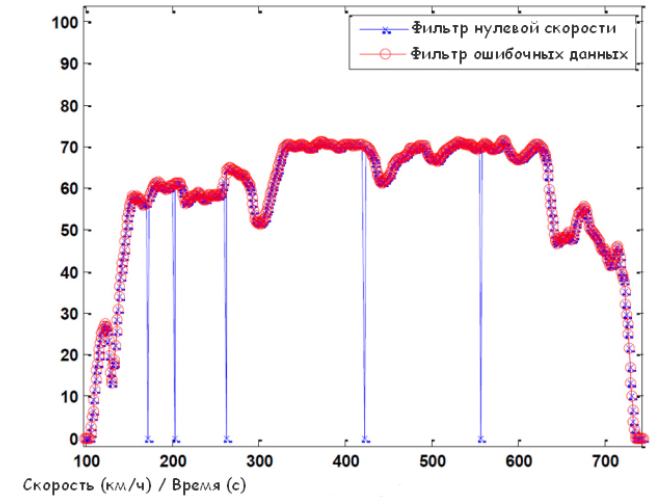


Рис. 3. Ложные результаты фильтра нулевой скорости для выборочных данных GPS о транспортном средстве

На этом этапе алгоритм фильтрации пытается исправить пробелы в связанном сигнале GPS со скоростью и временем, вызванные эффектами городских каньонов и внезапной потерей сигнала. Используя предоставленный пользователем предел временного промежутка, фильтр проверяет информацию о отметке времени, полученную из входного потока данных GPS, и пытается выполнить интерполяцию по любым промежуткам сигнала, продолжительность которых меньше указанного предела. Если длительность интервала сигнала меньше предельного значения, алгоритм генерирует монотонно увеличивающиеся по времени сигналы на основе частоты дискретизации базовых исходных данных. Чтобы сгенерировать «новые» данные о скорости, к вновь сгенерированной временной области применяется та же интерполированная подгонка кубической сплайновой кривой, которая использовалась на предыдущих этапах фильтрации. Если фильтр обнаруживает временной интервал, превышающий заданный пользователем предел, как показано на рисунке 4, фильтр увеличивает или уменьшает скорость транспортного средства в зависимости от того, находится ли точка данных на передней или задней границе промежутка в сигнале. Таким образом, рампа строится на основе заданных пользователем пределов ускорения/замедления. Конструкция пандусов особенно важна при попытке адаптировать полевые данные для приложений динамометрии шасси и моделирования транспортных средств. Устранение временных разрывов — это единственный шаг в процессе фильтрации, который добавит данные к существующему набору данных GPS (связанная информация о времени и скорости). Наряду с тестированием/экспериментальными приложениями восстановление сигнала, обеспечиваемое фильтром разрыва сигнала, также важно на эмпирическом уровне при изучении рабочего цикла транспортного средства, особенно при выполнении анализа работы в режиме реального времени.

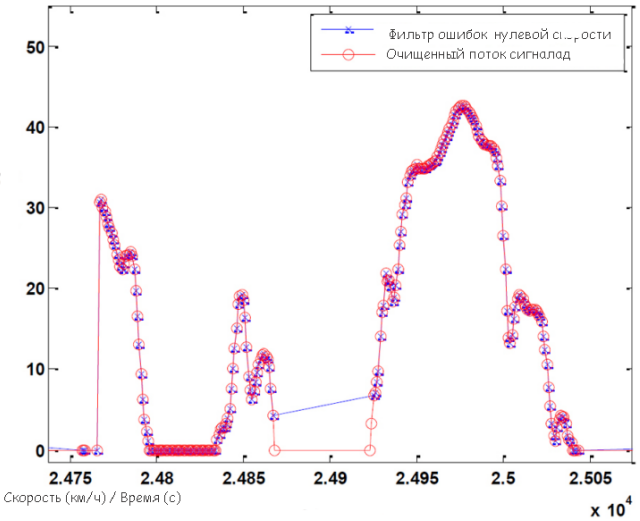


Рис. 4. Результаты фильтрации разрыва сигнала для выборочных данных GPS транспортного средства

Удаление шума из необработанных сигналов данных GPS и обеспечение дополнительных условий сглаживания являются неотъемлемой частью подготовки исходных данных для использования в приложениях для моделирования транспортного средства.

Например, устранение существующих переходных процессов сигнала и фонового шума особенно важно при проведении длительных испытаний шасси. Вместо того, чтобы пытаться следовать сильно изменяющейся траектории скорости и времени из-за компонентов шума сигнала, испытательный персонал может поддерживать более естественное поведение при движении и следовать плавному профилю,

более соответствующему типичному поведению при поездках. При моделировании транспортных средств удаление оставшегося сигнального шума позволяет избежать осложнений, возникающих при моделировании передовых стратегий управления транспортным средством. Зашумленные профили скорости генерируют неверную информацию о потребляемой мощности, что приводит к неточным расчетам состояния заряда и энергопотребления, что напрямую коррелирует с уменьшенными моделируемыми значениями экономии топлива.

Изучая результаты фильтрации для набора данных о транспортных средствах большой грузоподъемности, можно заметить, что по сравнению с приблизительной 20-процентной фильтрацией данных для данных о транспортных средствах малой грузоподъемности, данные о транспортных средствах большой грузоподъемности, рассмотренные в исследовании, показали фильтрацию в среднем менее 5 %. Когда игнорируются два этапа фильтрации, которые добавляют/удаляют данные, фильтрации подвергается в среднем менее 0,5 % собранных необработанных данных. При проведении сравнений анализа цикла движения как для необработанных, так и для отфильтрованных образцов GPS-цикла движения большегрузных транспортных средств, удаленные данные, по-видимому, оказывают минимальное влияние на качество необработанных данных о большегрузных транспортных средствах. В среднем максимальная и средняя скорость движения цикла снижаются, в то время как общая скорость цикла увеличивается, что предполагает устранение сбоев и скачков сигнала. Выполняя парные *t*-тесты для определения влияния фильтрации на показатели цикла привода, было обнаружено, что предлагаемый процесс фильтрации приводит к статистически значимым изменениям в средних значениях для каждого из рассмотренных показателей цикла привода. Однако, изучая различия в средних, медианных и стандартных отклонениях исследуемых показателей, оказывается, что, хотя изменения в среднем являются статистически значимыми, изменения в данных в целом минимальны.

Оценка экономии топлива

Поскольку особое внимание уделено повышению топливной экономичности автомобиля, способность правильно моделировать характеристики автомобиля по своей сути имеет решающее значение для цикла движения. В качестве окончательного сравнения сгенерированных циклов с собранной базой данных каждый сгенерированный цикл был прогнан через обычный симулятор для оценки экономии топлива. На рисунке 5 показано сравнение сгенерированной оценки экономии топлива в цикле со средней измеренной суточной экономией топлива, а также 95-процентый доверительный интервал для каждого среднего измеренного значения. Интервалы доверия были рассчитаны на основе количества точек данных в неделю и разницы между этими точками. На этом рисунке показано, что сгенерированный цикл попадает в доверительный интервал для каждого тестового города, что указывает на то, что сгенерированные города являются репрезентативными для типичных характеристик транспортных средств для каждого еженедельного набора данных [10].

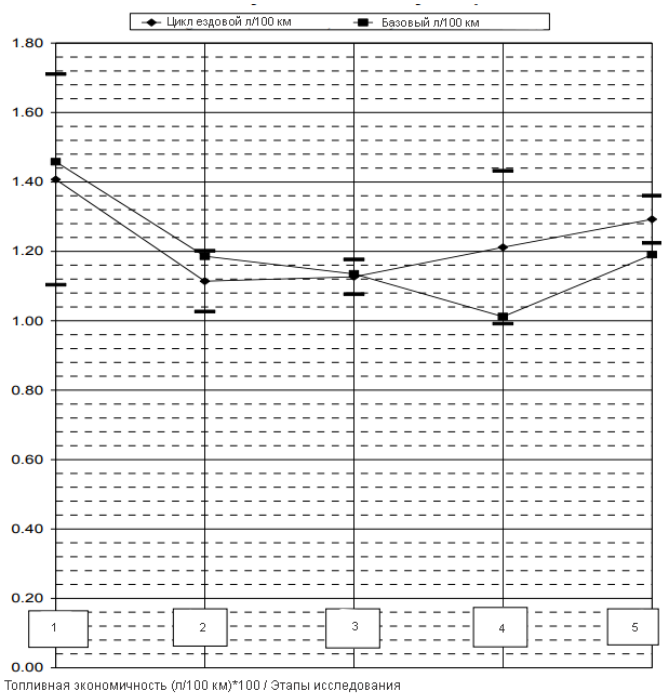


Рис. 5. Сравнение экономии топлива в генерируемых циклах с базовыми данными

Заключение

Основываясь на результатах данного исследования, можно сделать вывод, что предлагаемая фильтрация данных GPS адекватно удовлетворяет требованиям, предъявляемым к идеальному фильтру данных GPS. Исключая удаление ошибочных данных и добавление потерянных компонентов сигнала, предлагаемый процесс фильтрации минимально влияет на исходные данные, одновременно обеспечивая значительное улучшение качества данных, необходимых для расчета показателей цикла движения, используемых в рамках анализа цикла движения и приложений для моделирования транспортных средств.

Анализируя отфильтрованные результаты из наборов данных как для легких, так и для тяжелых транспортных средств, пробелы в сборе сигналов и лежащие в их основе проблемы со сбором данных являются гораздо более важным компонентом процесса фильтрации, нежели предполагалось ранее. Потеря сигнала GPS является наибольшим фактором, способствующим снижению качества данных в разбивке по точкам, и оказывает по меньшей мере на порядок большее влияние на процесс фильтрации, чем другие источники ошибок. Однако, изучив результаты анализа рабочего цикла на основе отфильтрованных и необработанных данных как для наборов данных о транспортных средствах большой грузоподъемности, было также показано, что устранение скачков данных является ключевым шагом в получении точных оценок скорости/ускорения транспортного средства. Основываясь на результатах этого исследования в сочетании с возросшим интересом к интеграции информации о высоте/уклоне дороги в инженерный анализ, в будущем рекомендуется разработать дополнительные методы фильтрации для восстановления собранной GPS информации о высоте. Объеди-

нение дополнительного измерения с процессом фильтрации должно предоставить дополнительные возможности фильтрации на основе логики, которые в настоящее время отсутствуют при анализе данных в одном измерении и оценке топливной эффективности.

Литература

1. Маняшин, А. В. Прогнозирование и планирование ресурсов на автомобильном транспорте с использованием информационных технологий. — Тюмень: ТюмГНГУ, 2015. — 145 с.

2. Маняшин, С. А. Автоматизация исследований режимов движения автомобилей в городе / С. А. Маняшин, А. В. Маняшин // Проблемы эксплуатации систем транспорта: Сборник материалов Всероссийской научно-практической конференции, посвященной 45-летию со дня основания Тюменского индустриального института им. Ленинского комсомола (Тюмень, Россия, 06 ноября 2008 г.) / отв. ред. В. И. Бауэр. — Тюмень: ТюмГНГУ, 2008. — С. 197–198.

3. Кувыкин, В. И. Анализ оптимальных решений в задачах нелинейного программирования / В. И. Кувыкин, Е. В. Кувыкина, М. Ю. Петухов // Вестник Нижегородского университета им. Н. И. Лобачевского. 2011. № 4 (5). С. 2285–2286.

4. Маняшин, С. А. Методика исследования режимов движения автомобилей в городских условиях / С. А. Маняшин, А. В. Маняшин // Проблемы эксплуатации и обслуживания транспортно-технологических машин: Материалы Международной научно-технической конференции (Тюмень, Россия, 14 апреля 2008 г.) / отв. ред. Н. С. Захаров. — Тюмень, ТюмГНГУ, 2008. — С. 126–128.

5. Вычисление низкотемпературных характеристик дизельного топлива с помощью искусственных нейронных сетей / С. Маринович, Т. Боланча, С. Юкич, [и др.] // Химия и технология топлив и масел. 2012. № 1 (569). С. 47–51.

6. Ясницкий, Л. Н. Интеллектуальные системы: Учебник. — Москва: Лаборатория знаний, 2016. — 221 с. — (Учебник для высшей школы).

7. A Dual Strategy for Controlling Energy Consumption and Air Pollution in China's Metropolis of Beijing / Y. Zhang, H. Wang, S. Liang, [et al.] // Energy. 2015. Vol. 81. Pp. 294–303. DOI: 10.1016/j.energy.2014.12.041.

8. CO2 Emissions from Fuel Combustion 2016. — Paris: International Energy Agency, 2016. — 533 p.
DOI: 10.1787/co2_fuel-2016-en.

9. Wang, J. Fuel Consumption Model for Conventional Diesel Buses / J. Wang, H. A. Rakha // Applied Energy. 2016. Vol. 170. Pp. 394–402. DOI: 10.1016/j.apenergy.2016.02.124.

10. Typaldos, P. Minimization of Fuel Consumption for Vehicle Trajectories / P. Typaldos, I. Papamichail, M. Papageorgiou // IEEE Transactions on Intelligent Transportation Systems. 2020. Vol. 21, Is. 4. Pp. 1716–1727.
DOI: 10.1109/TITS.2020.2972770.

DOI: 10.24412/2413-2527-2022-230-19-24

A Way to Filter GPS Data for a Driving Cycle. Investigation of the Fuel Efficiency of a Special Car

O. A. Durnitsyn
Industrial University of Tyumen
Tyumen, Russia
durnitsynoa@mail.ru

Abstract. **Global Positioning System data collection devices have proven to be useful tools for collecting real-world traffic data and statistics. The data collected by these devices provide valuable information when studying vehicle movement parameters. For vehicle modeling, this data is invaluable for analyzing fuel consumption and vehicle performance. Without reliable information about speed and time, the reliability of the obtained indicators for motion cycles, such as acceleration, power and distance, becomes questionable. This study examines some common sources of errors present in the collected raw GPS data, and presents a detailed filtering process designed to eliminate these problems, followed by plotting the impact of errors on the correctness of the calculation of fuel economy. The test data for severe operating conditions are analyzed. Graphical comparisons of raw and filtered cycles are presented, as well as statistical analysis to determine the effect of the proposed filtering process on the raw data. Finally, the article concludes with an assessment of the general advantages of filtering GPS data and presents the experience of calculating the fuel efficiency of the concrete mixer truck for which the cycle was built.**

Keywords: **GPS data, signal filtering, data filtering, fuel consumption, driving cycle.**

REFERENCES

1. Manyashin A. V. Forecasting and planning resources in road transport using information technology [Prognozirovanie i planirovanie resursov na avtomobil'nom transporte s ispol'zovaniem informatsionnykh tekhnologiy]. Tyumen, Tyumen State Oil and Gas University, 2015, 145 p.

2. Manyashin S. A., Manyashin A. V. Automation of Researches of Modes of Movement of Cars in the City [Avtomatizatsiya issledovaniy rezhimov dvizheniya avtomobiley v gorode], *Problems of Operation of Transport Systems: Collection of materials of the All-Russian Scientific and Practical Conference [Problemy ekspluatatsii sistem transporta: Sbornik materialov Vserossiyskoy nauchno-prakticheskoy konferentsii], Tyumen, Russia, November 06, 2008*. Tyumen, Tyumen State Oil and Gas University, 2008, Pp. 197–198.

3. Kuvykin V. I., Kuvykina E. V., Petukhov M. Yu. The Analysis of Optimal Solutions in Non Linear Programming Problems [Analiz optimal'nykh resheniy v zadachakh nelineynogo programmirovaniya], *Vestnik of Lobachevsky State University of Nizhni Novgorod [Vestnik Nizhegorodskogo universiteta imeni N. I. Lobachevskogo]*, 2011, No. 4 (5), Pp. 2285–2286.

4. Manyashin S. A., Manyashin A. V. Methodology for Research of Modes of Movement of Cars in Urban Conditions [Metodika issledovaniya rezhimov dvizheniya avtomobiley v gorodskikh usloviyakh], *Problems of Operation and Maintenance of Transport and Technological Machines: Proceedings of the International Scientific and Technical Conference [Problemy ekspluatatsii i obsluzhivaniya transportno-tekhnologicheskikh mashin: Materialy Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii], Tyumen, Russia, April 14, 2008*. Tyumen, Tyumen State Oil and Gas University, 2008, Pp. 126–128.

5. Marinovich S., Bolanca T., Ukich S., et al. Calculation of Low-Temperature Properties of Diesel Fuel Using Artificial Neural Networks [Vychislenie nizkotemperaturnykh kharakteristik dizel'nogo topliva s pomoshch'yu iskusstvennykh neyronnykh setey], *Chemistry and Technology of Fuels and Oils [Khimiya i tekhnologiya topliv i masel]*, 2012, No. 1 (569), Pp. 47–51.

6. Yasnitsky L. N. Intelligent Systems: Textbook [Intellektual'nye sistemy: Uchebnik]. Moscow, Laboratory of Knowledge, 2016, 221 p.

7. Zhang Y., Wang H., Liang S., et al. A Dual Strategy for Controlling Energy Consumption and Air Pollution in China's Metropolis of Beijing, *Energy*, 2015, Vol. 81, Pp. 294–303. DOI: 10.1016/j.energy.2014.12.041.

8. CO2 Emissions from Fuel Combustion 2016. Paris, International Energy Agency, 2016, 533 p. DOI: 10.1787/co2_fuel-2016-en.

9. Wang J., Rakha H. A. Fuel Consumption Model for Conventional Diesel Buses, *Applied Energy*, 2016, Vol. 170. Pp. 394–402. DOI: 10.1016/j.apenergy.2016.02.124.

10. Typaldos P., Papamichail I., Papageorgiou M. Minimization of Fuel Consumption for Vehicle Trajectories, *IEEE Transactions on Intelligent Transportation Systems*, 2020, Vol. 21, Is. 4, Pp. 1716–1727. DOI: 10.1109/TITS.2020.2972770.

DOI: 10.24412/2413-2527-2022-230-25-31

Автоматизация документооборота и бизнес-процессов с помощью ECM-системы Docsvision

М. С. Дмитриева, к.и.н. А. В. Забродин
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
marina-dm2000@mail.ru, teach-case@yandex.ru

Аннотация. **В рамках статьи рассматриваются системы управления электронным документооборотом в рабочих процессах делопроизводства. Показаны достоинства и недостатки ECM-систем, которые обеспечивают улучшение управления документами в целях оперативного взаимодействия между структурами внутри организации, конфиденциальность информации, а также сокращение временных и трудовых затрат. Рассмотрены преимущества последнего поколения систем управления электронным документооборотом на примере Docsvision.**

Ключевые слова: **Docsvision, информационная система, конфиденциальность, бизнес-процессы, ECM-система, SQL, электронный документооборот.**

Введение

Информационные технологии (ИТ) являются наиболее важной составляющей процесса использования информационных ресурсов общества. Одной из задач ИТ является обеспечение возможности применения высокотехнологичных технический средств и научных методов для реализации разнообразных информационных документационных процессов.

Современные технологии, позволяющие создавать, хранить, перерабатывать данные и информацию, обеспечивать эффективные способы представления информации, стали важным фактором конкурентоспособности и средством повышения эффективности управления всеми сферами общественной жизнедеятельности. Уровень информатизации является сегодня одним из главных факторов успешного развития всякого предприятия [1].

К настоящему времени ИТ в хранении и обработке информации прошли несколько эволюционных этапов. Первоначально в задачу управления документами входила исключительно простая автоматизация документооборота, то есть помощь делопроизводителям, но ни в коей мере не замена документных потоков, циркулирующих на предприятии. Затем появились индивидуальные для каждого предприятия первые ECM-системы (Content Management System — системы управления контентом), но они создавались внутренне и были не масштабируемы [2].

Развитие функциональности системы электронного документооборота (СЭД) привело к тому, что все больше документно-ориентированных процедур управления было автоматизировано и все больше областей повседневной деятельности компаний были покрыты их функционалом [3].

В нестоящее время на рынке существует значительное количество ECM-систем, такие как 1С:Документооборот, ELMA, Directum, Тезис, MS SharePoint и др. Все они, как правило, включают в себя типовой набор сервисов: настройку процессов бизнес-пользователям без программирования, установку на свой сервер, API, создание документа по шаблону, регистрацию из электронной почты, работу со словарями и справочниками, атрибутивный поиск документов и многое другое.

При всем многообразии положительных сторон, приведенных в таблице, у каждой из них есть свои недостатки. Например, в 1С: Документооборот и Microsoft SharePoint отсутствует обеспечение свободной маршрутизации документов и делегирование полномочий. Также 1С требует частого обновления, которое невозможно произвести без программиста, что несомненно требует дополнительных расходов. Невозможность работы из браузера снижает производительность из-за необходимости установки и постоянного обновления системы на каждый компьютер, использующий 1С.

Сравнительная таблица различных ECM-систем приведена на рисунке 1.

Частое обновление, низкая производительность, увеличенные трат — решение этих проблем обеспечивается за счет применения концепции CSP.

CSP (Content Service Platform) — это концепция построения системы управления корпоративным контентом, которая предлагает отказ от монолитной архитектуры ECM-систем в пользу платформы (базы) и набора сервисов [4].

При этом следует отметить, что неотъемлемой частью CSP-решений является использование технологии Low-code. Low-code — это подход к созданию приложений, при котором не нужно писать программный код в привычном понимании. При этом для создания и запуска ИТ-решения не требуется узкоспециализированных знаний. Платформы Low-code содержат готовые шаблоны и сценарии для оптимизации различных бизнес-процессов.

Low-Code-инструменты дают неограниченные возможности дорабатывать и кастомизировать систему с минимальным дополнительным программированием.

В целом концепция CSP включает в себя возможность разрабатывать и подключать сервисы и дополнительные модули как при внедрении системы, так и в процессе ее эксплуатации, быструю автоматизацию контуров новых процессов, отсутствие ограничений масштабирования, высокую производительность и отказоустойчивость [5]. Все это — ключевые характеристики и преимущества класса CSP-решений.

	Comindware	Directum	1С:Документооборот	DocVision	ELMA	Дело	Тезис	NauDoc	Евфрат
Настройка процессов бизнес-пользователями, без программирования	✓	✓	–	✓	✓	✓	–	✓	✓
Установка на свой сервер	✓	✓	✓	✓	✓	✓	✓	✓	✓
Web-клиент	✓	✓	–	✓	✓	✓	✓	✓	✓
Графовая база данных	✓	–	–	–	–	–	–	–	–
API	✓	✓	✓	✓	✓	✓	✓	✓	✓
Создание документа по шаблону	✓	✓	✓	✓	✓	✓	✓	✓	✓
Регистрация из электронной почты	✓	✓	✓	✓	✓	✓	✓	✓	✓
Работа со словарями и справочниками	✓	✓	✓	✓	✓	✓	✓	✓	✓
Возможность назначения ответственного исполнителя по документу	✓	–	–	–	✓	✓	✓	✓	✓
Обеспечение свободной маршрутизации документов	✓	✓	–	✓	✓	–	✓	✓	✓
Делегирование полномочий	✓	✓	–	✓	✓	✓	✓	✓	–
Рассылка уведомлений по электронной почте	✓	✓	✓	✓	✓	✓	✓	✓	✓
Атрибутивный поиск документов	✓	✓	✓	✓	✓	✓	✓	✓	✓
Поиск задач и поручений	✓	✓	✓	✓	✓	✓	✓	✓	✓
Средства мониторинга процессов в системе	✓	✓	✓	✓	✓	✓	✓	✓	✓
Соответствие требованиям импортозамещения РФ	✓	–	–	–	–	✓	–	–	–

Рис. 1. Сравнение ECM-систем

Docsvision — полнофункциональная CSP-платформа для автоматизации бизнес-процессов и задач обработки документов. Решения на базе платформы Docsvision помогают хранить весь цифровой контент организации, управлять документами и бизнес-процессами, объединять разрозненные бизнес-подразделения в едином «прозрачном» поле, а также соответствовать необходимым стандартам оформления документов.

Docsvision тесно интегрирована с платформой и приложениями Microsoft и разработанная на базе технологии .Net, представляющая собой кроссплатформенную среду выполнения приложений. Она позволяет приложениям работать на всех процессорах и на всех ОС семейства Windows (за исключением самых ранних).

В настоящее время актуальной версией является 5.5, в которой основные акценты в разработке были сделаны на масштабируемость, отказоустойчивость и удобную работу с документами в электронном архиве.

Архитектура платформы Docsvision

Технологически система Docsvision представляет собой трехуровневое приложение, где центральным объектом которого является карточка — совокупность описания структуры, соответствующих ей данных и одного или нескольких элементов управления ActiveX, реализующих пользовательский интерфейс [6].

Архитектура платформы Docsvision 5 полностью открыта. Все функции приложений Docsvision доступны с помощью программных интерфейсов, обеспечивающих беспрецедентную гибкость использования компонентов платформы для создания конкретных решений и интеграции с внешними системами. Помимо API платформа предоставляет инструменты (конструкторы) для визуального проектирования решений без использования программного кода (воплощение технологии Low-code).

Все приложения, создаваемые с использованием платформы Docsvision, обеспечивают:

• единые механизмы хранения и представления информации;

• единые механизмы разграничения прав доступа к электронным документам и действиям с ними;

• общие средства навигации и доступа к данным и организации рабочего пространства пользователя;

• общую структуру справочников;

• общие средства поиска и извлечения информации;

• единые средства маршрутизации и управления жизненным циклом документов;

• единые средства разработки и настройки;

• единую инфраструктуру управления и администрирования приложений;

Преимуществом данной архитектуры является оптимальное распределение функций между отдельными сервисами для обеспечения производительности системы, возможность независимого программного расширения каждого сервиса системы, а также возможность работы в разнообразном аппаратном окружении.

Общая архитектура платформы Docsvision приведена на рисунке 2.

При этом следует отметить, что архитектура платформы создавалась с учетом современных подходов к реализации промышленных систем, которые основаны на сокращении затрат, гибкости, клиентоориентированности, переходе от массового производства к потоку единичных изделий.

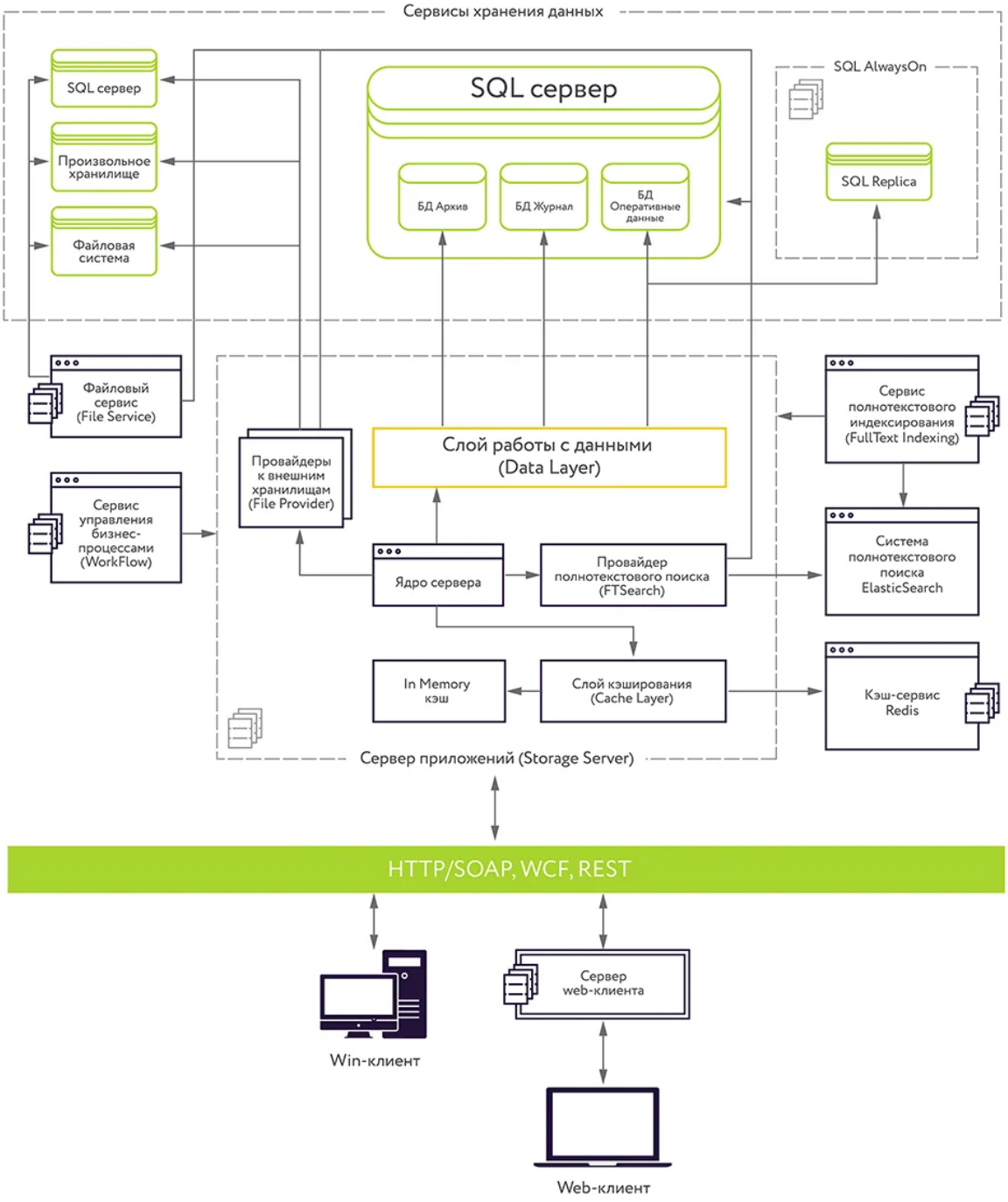


Рис. 2. Архитектура платформы Docsvision

В основе платформы Docsvision лежит современная трехзвенная клиент-серверная архитектура, спроектированная с учетом растущих потребностей в объемах обрабатываемых данных и количества вовлеченных в работу пользователей предприятия [7].

Архитектура «Клиент-Сервер» (также используются термины «сеть Клиент-Сервер» или «модель Клиент-Сервер») предусматривает разделение процессов предоставление услуг и отправки запросов на них на разных компьютерах в сети, каждый из которых выполняют свои задачи независимо от других.

В архитектуре «Клиент-Сервер» несколько компьютеров-клиентов (удалённые системы) посылают запросы и получают услуги от централизованной служебной машины — сервера (server – англ. «официант, обслуга»), которая также может называться хост-системой (host system, от host – англ. «хозяин», обычно гостиницы) [8].

Клиентская машина предоставляет пользователю так называемый «дружественный интерфейс» (user-friendly interface), чтобы облегчить его взаимодействие с сервером.

Исходя из представленной архитектуры, основным компонентом серверной архитектуры является сервер приложений (Storage Server). Посредством Storage Server осуществляется взаимодействие клиентских компонент системы и внешних приложений. Сервер приложений реализует прикладные сервисы доступа к данным, разграничение прав доступа, управление пользовательскими сессиями и т. д.

Для работы с данными сервер приложений обращается к сервисам хранения данных. Сервис баз данных обеспечивает хранение структурированных данных и обработку части бизнес-логики, а также включает возможность реализации на базе промышленной СУБД Microsoft SQL Server, PostgreSQL или Postgres Pro [9].

В состав стандартной поставки платформы Docsvision входят два приложения: «Делопроизводство» и «Управление процессами», реализующие соответствующую функциональность. Приложения содержат средства настройки, позволяющие создавать на их основе законченные решения для конкретных задач пользователя.

Дополнительные приложения, разработанные на базе платформы и частично использующие функциональность стандартных приложений, реализуют отдельные специфические задачи из области управления документами и бизнес-процессами.

Дополнительные модули расширяют технические возможности системы, например, модуль репликации между серверами, модули интеграции и шлюзы к внешним системам, веб-расширения доступа к системе через корпоративный портал, модуль архивирования, модуль расширенных отчетов и т. д. [10].

Windows-клиент и Web-клиент

На клиентском уровне архитектуры платформа делится на две части: Windows-клиент («толстый» клиент) и Web-клиент («тонкий» клиент).

Windows-клиент – приложение, которое устанавливается на компьютер, и обеспечивает полноценное рабочее место как для администратора системы, так и для рядового пользователя. «Толстый» клиент предлагает пользователю и администратору СЭД актуальный набор сценариев для организации работы в системе. Работу по ограниченным сценариям (например, исполнение заданий) можно организовать с помощью другого клиентского ПО Docsvision, например, клиента для Microsoft Outlook.

С увеличением запросов рынка к кроссплатформенности и мобильной работе более востребованным видом рабочего места стал выступать web-клиент Docsvision. Web-клиент обеспечивает пользователям максимальную мобильность, позволяя работать через интернет из любой точки мира и с любого устройства, адаптируясь под размер экрана и предоставляя удобный, понятный и привычный интерфейс [11].

Интерфейс «тонкого» клиента позволяет выполнять в системе все необходимые функции: работать с документами и заданиями, запускать процессы согласования и исполнения/ознакомления, осуществлять поиск по реквизитам и тексту документов, формировать и просматривать отчеты.

Основной интерфейс платформы Docsvision — web-клиент, который предоставляет пользователю доступ к системе через интернет-браузер. В «тонком» клиенте поддерживается работа со всеми объектами Docsvision: просмотр дерева папок и карточек, находящихся в папках; просмотр содержимого карточек и их группировку по определенным признакам; поиск карточек по заданным критериям, а также создание новых и редактирование уже существующих документов. Web-клиент работает во всех наиболее распространённых современных браузерах.

Web-клиент использует DHTML (Dynamic Hypertext Markup Language), HTTPRequest (класс, управляющий объектом Request, содержащем информацию о текущем запросе: его тип, параметры и их значения, класс «запроса к серверу») и элементы технологии AJAX (Asynchronous JavaScript and XML — «асинхронный JavaScript и XML»). Вся функциональность клиента в этом режиме исполняется системой автоматически с помощью языка JavaScript.

Компоненты платформы Docsvision

Модель приложения Docsvision включает следующие основные компоненты (рис. 3):

- Карточка — основной объект системы, который позволяет смоделировать любой прикладной объект приложения. Наиболее типичными объектами в приложениях СЭД являются карточки документов (входящий, исходящий, договор) и заданий (задание на исполнение, задание на согласование).
- Справочник — специальный вид карточки, существующий в системе в единственном экземпляре и позволяющий сохранять самую разнообразную справочную информацию, в частности, иерархически организованную.
- Процесс — объект, моделирующий последовательность выполнения действий в рамках приложения, группы взаимосвязанных приложений или на уровне системы. Процесс включает как ручные этапы (взаимодействия пользователя с карточками или внешними объектами) и автоматические этапы, реализуемые сервисом, для обеспечения логики обработки процесса.
- Папка — объект, обеспечивающий группировку карточек. В системе имеются обычные папки – группируют карточки, которые в них созданы, либо посредством физического размещения в них ярлыков на карточки (аналогично папкам файловой системы) — и виртуальные, которые группируют карточки по соответствию определенным критериям (результаты поиска).

Рис. 3. Компоненты, из которых формируется прикладное решение на платформе Docsvision

• Представление — групповое представление набора карточек в виде таблицы или набора строк. В ячейках таблицы отображаются атрибуты карточек, их агрегаты и результаты выполнения расчетных операций.

Любое приложение является комбинацией определенного набора из перечисленных выше объектов.

Преимущества платформы Docsvision

Платформа Docsvision решает широкий спектр задач, необходимые компаниям для обеспечения полномасштабного функционирования жизненного цикла документооборота.

1. Управление доступом. Система Docsvision обеспечивает возможность защиты информации на различных этапах формирования документа. Осуществление безопасности достигается с помощью разграничения прав, дополнительных папок, ролевой модели, учетных записей пользователей, мандатной безопасности, а также дополнительных средств защиты от утечек информации за пределы организации [12]. Доступ пользователя можно ограничивать на различных этапах согласования. Благодаря этому пользователь не сможет, например, удалить отправленный документ или изменить его на этапе подписания. Эта особенность позволяет сохранить целостность и конфиденциальность данных [13].

2. Гибкость расширения функционала. Законченные решения для конкретных заказчиков создаются партнерами Docsvision при внедрении системы. Эти решения могут выполняться в виде настроек приложений Docsvision без программирования, или путем программирования на платформе. В каждом решении возможно сочетание этих двух подходов в нужном соотношении, что позволяет создать любую обработку события, например при нажатии на кнопку.

3. Взаимодействие клиентской и серверной частей, доступ клиента к базе данных осуществляется через web-сервер на базе Internet Information Services операционной системы Microsoft Windows Server. При этом используется либо сочетание протоколов SOAP и HTTP, позволяющее осуществлять взаимодействие не только по локальной сети, но и через Интернет (возможность осуществлять передачу данных клиентам, не входящим в закрытую сеть, ограничив при этом права доступа), либо служба Microsoft Windows Server, предоставляющая доступ к серверу внутри локальной сети по более быстрому именованному транспортному каналу (named pipe). Благодаря взаимодействию можно использовать одновременно и Windows-клиент, и Web-клиент, не боясь, что данные затеряются.

4. Не требует обучения. Интерфейс очень прост в использовании, поэтому нет необходимости какого-либо дополнительного обучения для работы на платформе Docsvision.

5. При открытии карточки некоторые поля могут быть заполнены автоматически, такие как дата создания, автор и др. После заполнения карточки ее можно отправить на согласование руководителю, также можно разослать задания по этой карточке назначенным сотрудникам. Можно присвоить документу состояние «Закрыт», тогда он будет доступен только для чтения. Таким образом, документ проходит весь цикл документа, настроенный в конструкторе состояний.

6. В дополнение к возможностям системы Docsvision можно использовать сторонние продукты, которые дают возможность контролировать документооборот за пределами организации и позволяют настроить такие сценарии, как запрет печати документов из системы, даты, в рамках которых действуют полномочия сотрудника на работу с документом, исключается возможность отправки доку-мента по электронной почте. Одним из таких продуктов является служба управления правами Microsoft Windows Rights Management Server, реализованная в виде службы Windows Server 2008. Таким образом, обеспечить защиту от потери данных в рамках работы с документами внутри организации, можно с помощью возможностей системы Docsvision, а для исключения утечек информации за пределы компании, необходимо использовать сторонние продукты и приложения [14].

Заключение

Платформа Docsvision — эффективное решение, которое позволяет организовать слаженную работу с электронными документами. Стремительный рост и развитие современных технологий приводят к тому, что ECM-системы становятся неотъемлемой частью любого предприятия. При этом объем задач, которые она должна решать, становится шире. Целью развития системы является создание инструмента, обеспечивающего решение всего разнообразия задач обработки документов для организаций различного масштаба – от средних предприятий до крупнейших

корпораций и госструктур. Docsvision спроектирована с учетом этих тенденций. Данная система соответствует международным стандартам делопроизводства, обеспечивает широким функционалом и сохраняет безопасность данных, а дружественный интерфейс Docsvision обеспечивает удобство использования платформы без предварительного обучения и знаний программирования.

Таким образом, платформа Docsvision включает в себя все современные решения, от классического делопроизводства до организации полноценного цифрового управления.

Литература

1. Архитектура информационных систем: Учебник для студентов вузов / Б. Я. Советов, А. И. Водяхо, В. А. Дубенецкий, В. В. Цехановский. — Москва: Издательский центр «Академия», 2012. — 288 с. — (Высшее профессиональное образование. Бакалавриат).

2. Концепция «Enterprise Content Management» (ECM) // Справочник секретаря и офис-менеджера. Электронный журнал. — 2016. — 30 июня. URL: http://www.sekretariat.ru/article/210499-qqe-16-m6-kontseptsiya-enterprise-content-management-esm (дата обращения 16.05.2022).

3. Делопроизводство (Организация и технологии документационного управления) / под ред. Т. В. Кузнецовой. — Москва: ЮНИТИ-ДАНА, 2001. — 359 с.

4. Баласанян, В. Э. Концепция системы автоматизации отечественного документооборота // Открытые системы. СУБД. 1997. № 1 (21). С. 42–44.

5. Байкова, И. Ю. Документооборот и делопроизводство. Как организовать работу с документами. — Москва: Эксмо, 2010. — 287 с. — (Офисная антология).

6. Чеппел, Д. Технологии ActiveX и OLE = Understanding ActiveX and OLE / Пер. с англ. Д. Г. Новоселова. — Москва: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1997. — 320 с. — (Стратегия программирования).

7. Дрождин, В. В. Эволюция архитектуры информационных систем / В. В. Дрождин, Р. Е. Зинченко // Программные продукты и системы. 2010. № 4. С. 59–63.

8. Архитектура «Клиент-Сервер» // Блог компании «Ителон». — 2021. — 17 августа. URL: http://itelon.ru/blog/arkhitektura-klient-server (дата обращения 16.05.2022).

9. Описание CSP-системы Docsvision 5.5. URL: http://docsvision.com/besplatnye-materialy/polnoe-opisanie-platformy-docsvision-55 (дата обращения 16.05.2022).

10. Назначение системы DocsVision // Сайт компании «Интерфейс». — 2008. — 23 июля. URL: http://www.interface.ru/home.asp?artid=16890 (дата обращения 16.05.2022).

11. Рябов, В. А. Современные веб-технологии / В. А. Рябов, А. И. Несвижский. — 2-е изд., испр. — Москва: НОУ «ИНТУИТ», 2016. — 1080 с.

12. Теория информационных процессов и систем / под ред. Б. Я. Советова. — Москва: Издательский центр «Академия», 2010. — 432 с. — (Университетский учебник. Серия «Прикладная математика и информатика»).

13. Конфиденциальное делопроизводство и защищенный электронный документооборот / Н. Н. Куняев, А. С. Демушкин, Т. В. Кондрашова, А. Г. Фабричнов; под общ. ред. Н. Н. Куняева. — 2-е изд., перераб. и доп. — Москва: Логос, 2020. — 500 с. — (Новая университетская библиотека).

14. Макаревич, Е. Защищенный документооборот DocsVision // Storage News. 2010. № 4 (44). С. 26–27.

DOI: 10.24412/2413-2527-2022-230-25-31

Automation of Document-Flow and Business Processes Using the ECM-System Docsvision

M. S. Dmitrieva, PhD A. V. Zabrodin
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
marina-dm2000@mail.ru , teach-case@yandex.ru

Abstract. **Within the framework of the article, electronic document management systems in office workflows are considered. The advantages and disadvantages of ECM systems are shown, which provide improved document management for the purpose of operational interaction between structures within the organization, confidentiality of information, as well as reduction of time and labor costs. The advantages of the latest generation of electronic document management systems are considered on the example of Docsvision.**

Keywords: **Docsvision, information system, confidentiality, business processes, ECM system, SQL, electronic document management.**

REFERENCES

1. Sovetov B. Ya., Vodyakho A. I., Dubenetskiy V. A., Tsekhanovskiy V. V. Architecture of information systems: Textbook for university [Arkhitektura informatsionnykh sistem: Uchebnik dlya studentov vuzov]. Moscow, Publishing Center «Academy», 2012, 288 p.

2. The Concept of Enterprise Content Management (ECM) [Kontseptsiya «Enterprise Content Management» (ESM)], *Directory of Secretary and Office Manager. Electronic Journal [Spravochnik sekretarya i ofis-menedzhera. Elektronnyy zhurnal]*. Published online at June 30, 2016. Available at: http://www.sekretariat.ru/article/210499-qqe-16-m6-kontseptsiya-enterprise-content-management-esm (accessed 16 May 2022).

3. Kuznetsova T. V. (ed.) Office work (Organization and technologies of document management) [Deloproizvodstvo (Organizatsiya i tekhnologii dokumentatsionnogo upravleniya)]. Moscow, Unity-Dana, 2001, 359 p.

4. Balasanyan V. E. The Concept of the Domestic Document Management Automation System [Kontseptsiya sistemy avtomatizatsii otechestvennogo dokumentooborota], *Open Systems. DBMS [Otkrytye sistemy. SUBD]*, 1997, No. 1 (21), Pp. 42–44.

5. Baykova I. Yu. Document management and office work. How to organize work with documents [Dokumentooborot i deloproizvodstvo. Kak organizovat rabotu s dokumentami]. Moscow, The Eksmo Publishing House, 2010, 287 p.

6. Chappel D. Understanding ActiveX and OLE [Tekhnologii ActiveX i OLE]. Moscow, Publishing Department «Russian Edition» Channel Trading Ltd., 1997, 320 p.

7. Drozhdin V. V., Zinchenko R. E. Evolution of Information Systems Architecture [Evolyutsiya arkhitektury informatsionnykh sistem], *Software and Systems [Programmnye produkty i sistemy]*, 2010, No. 4, Pp. 59–63.

8. Client-Server Architecture [Arkhitektura «Klient-Server»], *Itelon*. Published online at August 17, 2021. Available at: http://itelon.ru/blog/arkhitektura-klient-server (accessed 16 May 2022).

9. Description of the CSP-System Docsvision 5.5 [Opisanie CSP-sistemy Docsvision 5.5]. Available at: http://docsvision.com/besplatnye-materialy/polnoe-opisanie-platformy-docsvision-55 (accessed 16 May 2022).

10. Purpose of the DocsVision System [Naznachenie sistemy DocsVision], *Interface Ltd*. Published online at July 23, 2008. Available at: http://www.interface.ru/home.asp?artid=16890 (accessed 16 May 2022).

11. Ryabov V. A., Nesvizhskiy A. I. Modern web technologies [Sovremennye veb-tekhnologii]. Moscow, National Open University INTUIT, 2016, 1080 p.

12. Sovetov B. Ya. (ed.) Theory of information processes and systems [Teoriya informatsionnykh protsessov i sistem]. Moscow, Publishing Center «Academy», 2010, 432 p.

13. Kunyaev N. N., Demushkin A. S., Kondrashova T. V., Fabrichnov A. G. Confidential office work and secure electronic document management [Konfidentsialnoe deloproizvodstvo i zashchishchennyy elektronnyy dokumentooborot]. Moscow, Logos, 2020, 500 p.

14. Makarevich E. Secure Document Management DocsVision [Zashchishchennyy dokumentooborot DocsVision], *Storage News*, 2010, No. 4 (44), Pp. 26–27.

DOI: 10.24412/2413-2527-2022-230-32-40

Методы построения кластерных космических систем на основе многоагентных технологий

к.т.н. В. А. Гончаренко
Петербургский государственный университет
путей сообщения Императора Александра I,
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия
vlango@mail.ru

к.т.н. В. С. Забузов
Военно-космическая академия
имени А. Ф. Можайского
Санкт-Петербург, Россия
zzvvss80@mail.ru

Аннотация. **Предложены основные принципы и методы построения системы малых космических аппаратов в виде кластера на основе многоагентных технологий. Рассмотрены основные подходы к повышению устойчивости функционирования систем малых космических аппаратов. Предложены архитектурные решения по построению многоагентной системы малых космических аппаратов, в основе которой лежат концепция кластеризации бортовых вычислительных систем и многофункциональная типология агентов.**

Ключевые слова: **многоагентная система, кластер спутников, малый космический аппарат, интеллектуальный агент, устойчивость функционирования, живучесть, распределенная система, бортовая вычислительная система.**

Введение

За последнее десятилетие благодаря успехам в микроэлектронике и ракетно-космической отрасли резко возросла популярность индустрии малых космических аппаратов (МКА). МКА применяются для решения различных задач наблюдения и связи, в том числе мобильной спутниковой связи, экологического мониторинга, картографирования территорий, поиска полезных ископаемых, контроля и оценки последствий чрезвычайных ситуаций и т. д.

В отличие больших и средних спутников, стоимость запуска и эксплуатации которых достаточно велика, малые спутники, используемые на низких и геостационарных орбитах, имеют хорошие перспективы развития нового направления спутниковой техники.

Согласно сложившейся классификации [1, 2] малыми космическими аппаратами называют спутники с массой менее 500 кг, для которых установлена международная классификация: *мини-спутники* (100–500 кг), *микроспутники* (10–100 кг), *наноспутники* (1–10 кг), *пикоспутники* (100 г–1 кг), *фемтоспутники* (до 100 г). Также часто спутники весом менее 10 кг в публикациях называют *сверхмалыми космическими аппаратами* (СМКА) [3, 4].

Основная миссия МКА — удешевление доступа в космическое пространство за счет их низкой стоимости [2]. Низкая стоимость услуг МКА обусловлена развертыванием серийного производства, низкой ценой средств запуска и самих МКА, сокращением времени реализации проекта от заказа до вывода МКА на орбиту, участием большого числа частных инвесторов в реализации проектов, возможностью резервирования, накопления на запасных орбитах МКА и гибкого изменения конфигурации космических систем на базе МКА.

В перспективе количество МКА на околоземных орбитах будет стремительно расти. В последние годы налажено производство наноспутников (СМКА), которые имеют все шансы занять свою нишу на космическом рынке [5]. В частности, уже сейчас успешно реализуется и активно развивается проект наноспутников формата CubeSat с размерами модуля 100×100×100 мм и массой 1–3 кг [2].

Для развертывания и пополнения орбитальных систем МКА используется так называемая *групповая схема* выведения на орбиту. При этом существует возможность создания *сетевых орбитальных группировок*, включающих десятки МКА. Это требует коренных изменений в принципах построения и организации управления системой МКА. Образующиеся при этом целые кластеры (или формации) МКА различного назначения будут постепенно вытеснять средние и большие КА за счет более низкой стоимости услуг МКА и быстрого внедрения новых технологий [6, 7]. Так, компания SpaceX с 2018 года успешно запускает сразу по 60 микроспутников Starlink и к июню 2022 года вывела на орбиту уже более 2 700 МКА. Около 250 спутников уже сведены с орбиты.

Использование кластерного принципа построения орбитальных группировок МКА подразумевает распределение целевой аппаратуры по нескольким МКА. Стало возможным формирование группы спутников, движущихся по орбите на небольшом удалении друг от друга и решающих одну задачу. Принципиальной особенностью группового полета является *автономная навигация* отдельного спутника в составе группы и *управление их взаимным относительным положением* в группе. Необходимость управления многочисленной группировкой требует смещения акцентов в системе управления с управления одним МКА на управление группировкой как сложной распределенной системой, с одновременным повышением степени автономности функционирования, составляющих группировку МКА [8].

В статье рассматриваются вопросы кластерного построения системы малых космических аппаратов для организации распределенной обработки информации в орбитальном и наземном сегментах на основе многоагентных технологий с целью обеспечения устойчивого функционирования космической системы [4, 9, 10].

Основы построения перспективных кластерных систем малых космических аппаратов

В развитии космических систем прослеживается тенденция, аналогичная для развития средств вычислительной техники — переход от громоздких больших ЭВМ к мини- и микроЭВМ, первоначально значительно уступавшим в производительности и функциональных возможностях,

затем постепенное наращивание и реализация в персональных ЭВМ современных технологий и достижение показателей, значительно превосходящих бывшие мейнфреймы. Аналогичный процесс наблюдается в космической индустрии. На смену тяжелым КА приходят мини-, микро-, наноспутники.

Отличительными особенностями проектов МКА являются: малая масса и габариты, использование серийных базовых компонентов МКА, дешевые средства выведения (легкие ракеты-носители в качестве дополнительной нагрузки), упрощенные унифицированные средства приема-передачи информации и т. д. В дальнейшем с ростом рынка услуг в области малой космической техники развитие космических систем МКА представляется по пути *интеллектуализации* орбитального и наземного сегментов [11]. Построение системы МКА, в отличие от традиционных систем, будет основываться на распределенном управлении группой МКА на основе принципов автономности, рациональности и адаптивности [9].

Построение наземного сегмента космической системы. В настоящее время управление осуществляется каждым КА в отдельности. В существующей орбитальной группировке насчитывается более 160 спутников, и наземному комплексу управления (НКУ) приходится проводить до 2 тысяч сеансов управления ежесуточно. Однако дальнейший рост группировки с учетом ограничений по наращиванию ресурсов управления создает вероятность коллапса. Новая же саморегулируемая технология позволит реализовать отечественные проекты многоспутниковых орбитальных группировок численностью от нескольких сотен до тысяч КА для обеспечения навигации, связи, дистанционного зондирования Земли и других функций [12].

Для решения задач предоставления и доставки услуг потребителю наземный сегмент системы МКА должен иметь распределенную сетевую инфраструктуру центров и пунктов управления (в том числе мобильных), пунктов приема и обработки пользовательской информации, средств доступа пользователей к услугам и информации со спутниковых систем. Примером такой системы может служить Единая территориально-распределенная информационная система дистанционного зондирования Земли (ЕТРИС ДЗЗ), созданная для унификации технических средств и программного обеспечения. Применение ЕТРИС ДЗЗ позволяет изменить парадигму создания перспективных космических систем путем отказа от разработки наземных комплексов приема, обработки и распространения спутниковой информации отдельно для каждой системы [13].

Центр управления полетом (ЦУП) НКУ должен быть способен одновременно управлять несколькими МКА, обрабатывая поступающую от них телеметрическую и служебную информацию и выдавая на борт командно-программную информацию в автоматическом режиме. Очевидно, что выполнить это возможно с помощью распределенного интеллектуального управления [11]. Учитывая существующую тенденцию к сокращению пунктов управления (вплоть до однопунктовой технологии управления), можно предположить, что для управления кластером МКА потребуется сеть интеллектуальных спутников-ретрансляторов, которые, по сути, будут реализовывать функции ЦУПа и радиотехнических систем, но размещаясь не на поверхности Земли, а в низкоорбитальном околоземном пространстве [14]. Как и для любой сложной информационной системы, встает вопрос о надежности и безопасности функционирования такой сети, своевременности предоставления услуг, разграничении доступа различных категорий пользователей к полезной информации. При этом для России необходимо еще решить вопрос об использовании получаемой информации в различных отраслях и в коммерческих целях, о возможности приема и обработки информации в центрах с МКА различных государств. Здесь наиболее отчетливо выступает задача разграничения доступа к информации различным категориям пользователей.

Для реализации средств доступа пользователей к космической информации возможно использовать инфраструктуру уже существующей глобальной сети Internet и базовые протоколы доступа к информации. При этом необходима разработка специальных интерфейсов доступа к пользовательской информации как на основе Web-технологий, так и современных сетевых технологий. Возможно также комплексное решение задач автоматизированного управления МКА и получения целевой информации с МКА с помощью интеграции функций в едином информационно-управляющем комплексе [14].

Построение орбитального сегмента системы МКА. Для группировок спутников возможно построение в виде *созвездия* (*Satellite Constellation*), когда обеспечивается равномерное покрытие Земли (как для КА GPS, GLONASS, Iridium), или в виде *формации* (*Formation Flying*) — единой сетевой спутниковой системы. Под формацией спутников понимают группу спутников, летящих на сравнительно близком взаимном расстоянии и работающих совместно над выполнением одной и той же задачи. МКА выводятся на орбиту в составе группы. Дальнейшее их управление также обеспечивается в составе формации. В зависимости от взаимного положения спутников и их ролей различают следующие типы формаций: трейлинги, рои, кластеры, фрагментированные спутники [15].

Под *трейлингом* (*trailing formation*) понимается группа спутников, вращающихся по одной и той же орбите и следующих друг за другом через определенный интервал времени.

Рой спутников (*swarm*) — группа идентичных спутников, объединенных для достижения одной цели без фиксации взаимного положения [16].

Под *кластером* (*cluster*) понимается совокупность КА различного целевого назначения, не только совместно выполняющих общую задачу, но и управляемых как единый космический объект и воспринимаемых потребителем как единое целое [7]. В кластер могут входить спутник-лидер, ведомые спутники и спутники-ретрансляторы.

Фрагментированный спутник (*fractionated spacecraft*) — это спутниковая архитектура, в которой функциональные возможности обычного монолитного КА распределены между несколькими модулями, которые взаимодействуют по беспроводным каналам связи [15]. В отличие от других типов формаций модули фрагментированного КА в значительной степени неоднородны и выполняют различные функции (соответствующие, например, различным подсистемам монолитного спутника).

Далее в статье кластером будем называть формацию МКА без уточнения ее типа, хотя в основном подразумевается кластерная структура.

Для согласованности действий МКА в составе системы требуются некоторые бортовые вычислительные и коммуникационные средства. По аналогии с распределенными вычислительными системами необходимо создать распределенную космическую систему с широким использованием интеллектуальных спутников-ретрансляторов для передачи управляющей информации.

Важным вопросом является автоматизация функций управления полетом кластера МКА. Существующие тенденции в развитии космических технологий говорят о возрастании автономности функционирования МКА, постепенном уменьшении участия измерительных пунктов в управлении [9]. С увеличением интеллектуальности и автономности МКА возможен переход от централизованной к распределенной технологии управления.

Методы построения устойчивых к воздействиям систем малых космических аппаратов на основе многоагентных технологий

Одной из основных проблем организации управления МКА является обеспечение устойчивости функционирования системы МКА в условиях возмущающих воздействий [17, 18].

Под *устойчивостью функционирования* будем понимать способность космической системы выполнять свои функции в условиях возмущающих воздействий различного рода, а также восстанавливать свои характеристики при выводе из устойчивого состояния [10, 18].

В общем случае, возмущающие воздействия могут носить механический, термический, электромагнитный, радиационный и информационный характер.

Понятие устойчивости функционирования системы объединяет такие более узкие понятия, как надежность, живучесть, помехоустойчивость, а также отказоустойчивость, робастность, радиационная стойкость, устойчивость к информационным воздействиям и др.

Оценить устойчивость такой сетевой структуры к различным деструктивным воздействиям можно на основе графовой модели с учетом связности маршрутов между спутниками [19].

Основными проблемами обеспечения устойчивости функционирования системы сверхмалых космических аппаратов являются [10]:

– возможность потери управления одним из объектов из-за сложности задачи распределенного управления;

– возможность потери управления орбитальной группировкой в условиях использования однопунктовой технологии и при отказе элементов НКУ;

– возможность перегрузки средств управления МКА при росте их количества, увеличении информационных потоков;

– увеличение последствий от действий человека при вводе ошибочных данных или неправильных действий по управлению КА.

Для обеспечения устойчивости функционирования НКУ и его компонентов могут быть использованы общие методы повышения устойчивости:

– методы повышения качества компонентов системы;

– структурные методы;

– различные методы и формы избыточности;

– эксплуатационные методы.

Основными направлениями повышения и обеспечения устойчивости функционирования системы МКА являются:

– обеспечение высокой отказоустойчивости элементов НКУ;

– осуществление мониторинга и управления сетевой инфраструктурой орбитальных и наземных средств управления космической системой;

– обеспечение помехоустойчивости каналов связи;

– применение современных технологий распределенной адаптивной диспетчеризации и управления информационными потоками от МКА-абонентов;

– повышение автономности МКА;

– повышение интеллектуальности управления МКА;

– рациональное распределение ресурсов между компонентами БКУ и НКУ для повышения устойчивости всей системы;

– внедрение адаптивных архитектур управляющих вычислительных систем;

– применение многоагентных технологий управления;

– применение различных видов и уровней резервирования в орбитальном и наземном сегментах космической системы МКА.

Одной из перспективных технологий, которая может быть положена в основу управления и получения целевой информации в системах МКА, является *технология многоагентных систем*, возникшая на стыке искусственного интеллекта и телекоммуникационных сетей, и основанная на принципах распределенного интеллектуального управления [14, 20].

Для управления системой МКА используется многоагентная концепция, предполагающая возможность одновременного контроля за большим количеством меняющихся источников информации в автоматическом режиме.

Интеллектуальный агент — это программная система, обладающая свойствами автономности, коммуникативности, реактивности, активности, интеллектуальности (обучаемости) [21]. Каждый агент — это процесс, обладающий определенной частью знаний об объекте управления и возможностью обмениваться этими знаниями с другими агентами. В зависимости от типа агент может поддерживать и интерфейс с пользователем.

Каждый тип агента предназначен для выполнения специальной функции. По существу интеллектуальные программные агенты должны выполнять задачи, делегируемые им пользователями. В зависимости от конкретного агента они могут размещаться на серверах, отдельных ЭВМ или в корпоративных сетях.

Многоагентная система (МАС) — это совокупность агентов, распределенных в сети и способных перемещаться по ней в поисках необходимых данных, знаний и процедур, взаимодействовать и кооперироваться друг с другом при выработке решений [14]. Цель агентов — удовлетворение различных информационных и вычислительных потребностей конечных пользователей, например: маршрутизация пакетов данных в сети, поиск и обобщение необходимой информации. МАС должна обладать такими свойствами, как распределенность в сети, независимость, размножаемость взаимодействующих агентов.

МАС можно представить, как многокомпонентную систему, состоящую из агентов со специфицированным интерфейсом. МАС можно описать следующей четверкой:

$$S_{ma} =< A_n, E, R, F >,$$

где A_n — множество агентов n типов;
E — глобальная телекоммуникационная среда, в которой происходит взаимодействие агентов;
R — множество отношений между агентами в коммуникационной среде;
F — множество функций (действий), выполняемых агентами.

МАС может содержать несколько однотипных или разнотипных агентов, которые могут иметь общие и/или различные цели, могут быть распределенными по компьютерной сети, могут быть написаны на различных языках программирования и работать на различных операционных платформах.

В качестве средств доступа пользователей к услугам и информации, полученной с МКА, используется система интеллектуальных агентов, каждый из которых способен предоставлять пользователю собранную всеми агентами информацию, временно выполняя роль менеджера системы мониторинга (рис. 1).

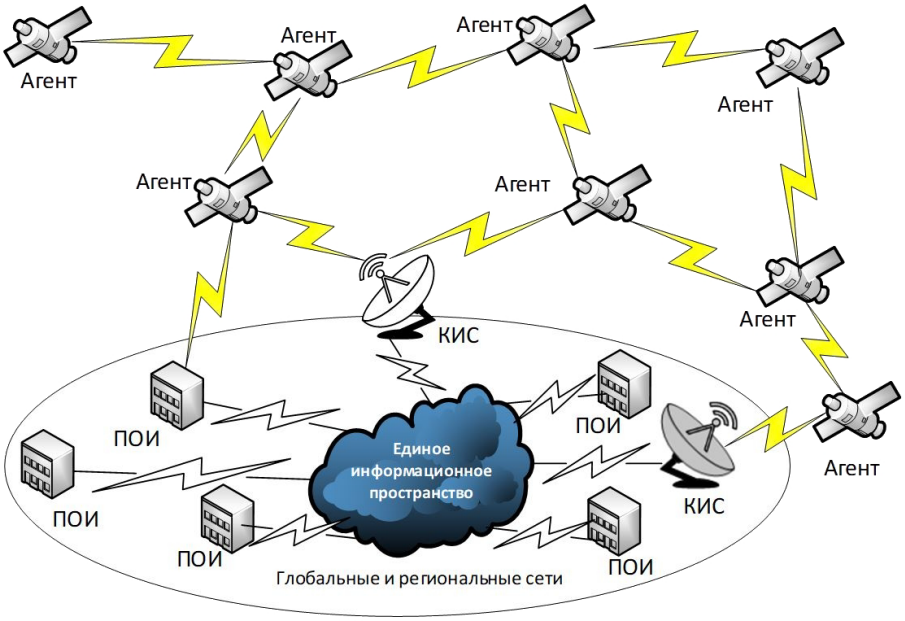


Рис. 1. Построение системы МКА

Ресурсы распределенных информационных систем будут доступны пользователям, обладающим средствами удаленного доступа через сеть Интернет, посредством интеллектуальных агентов. Агенты способны перемещаться по узлам сети и менять свою численность в зависимости от разных условий.

Взаимодействие МКА между собой также может быть построено по многоагентной технологии. В перспективе рассматривается разработка агентских программ, способных находить других агентов для решения поставленных задач. Кооперативное взаимодействие агентов должно быть динамическим и способным к адаптации в зависимости от текущей обстановки. Предполагается, что заявки на информацию, которую должны собирать МКА, могут поступать в любой наземный пункт НКУ, а общий поток заявок складывается из дискретных потоков заявок, поступающих в различные наземные пункты НКУ [20]. Заявки на сбор информации в кластер МКА также могут поступать от любого наземного пункта НКУ. Исполнителями заявок являются МКА кластера. Любой МКА может быть одновременно участником одной или нескольких команд, то есть может участвовать параллельно в исполнении одной или нескольких заявок на перекрывающихся интервалах времени. В процессе функционирования спутники кластера взаимодействуют между собой и с наземными пунктами НКУ, используя существующие коммуникационные каналы. Множество этих каналов между спутниками образует динамическую коммуникационную сеть. Для поддержки взаимодействия МКА кластера между собой и их взаимодействия с наземными пунктами НКУ необходима специальная программно-коммуникационная инфраструктура, в задачи которой входит решение задачи маршрутизации сообщений, задачи поиска спутников, обладающих заданными способностями по сбору и передаче информации. Заявка может пересылаться в космическую систему МКА через любой спутник. Аналогично полагается, что результат выполнения заявки также может доставляться любому наземному пункту. Каждый наземный пункт и каждый спутник кластера представлен *программным агентом*, который представляет соответствующий объект системы во всех процессах распределенного взаимодействия наземных пунктов и МКА орбитального кластера.

В перспективе получат свое развитие и многоагентные робототехнические системы [22–24]. Унификация и стандартизация аппаратных средств и программного обеспечения бортовых систем управления роботами, как активных элементов МАС, становится тем важнейшим фактором, который будет определять возможность и эффективность

обеспечения их информационно-логического взаимодействия и совместимости. Обеспечение автономности робота (как самостоятельного элемента МАС, априорно ориентируемого на работу в условиях неопределенности) предполагает наличие интеллектуальной бортовой системы управления, имеющей иерархическую структуру и реализующей весь спектр необходимых функций на основе комплексного применения современных технологий обработки знаний [22]. Кроме того, для автономных космических робототехнических систем (РТС) актуальна и проблема живучести [24]. Для обеспечения живучести такая РТС должна обладать дополнительными свойствами многоразовости, само- и взаимовосстанавливаемости, резервирования, модульности, приспособляемости.

Исходя из предложенных методов построения и реализации наземного и орбитального сегментов системы МКА, возникает ряд новых важных научных задач:

– разработка и оптимизация структуры наземного и орбитального сегментов системы МКА;

– разработка методов маршрутизации информационных сообщений в глобальной системе спутниковой связи и ретрансляции данных на основе МКА;

– организация трафика сообщений наземных абонентов — потребителей информации;

– разработка методов оптимизации орбитальной группировки с точки зрения обеспечения максимальной пропускной способности, надежности каналов доставки сообщения, минимизации затрат на управление орбитальными группировками.

Без правильно организованного управления сложные системы, к которым относится и кластер спутников, постепенно деградируют как в плане эффективности, так и в плане устойчивости. Поэтому каждый объект системы должен обладать свойством управляемости. Под *управляемостью* понимается возможность перевести систему с помощью вектора управления из произвольного начального состояния в произвольное конечное состояние за ограниченное время. Для управления объектами создается распределенная *интеллектуальная система управления* информационной инфраструктурой. Интеллектуальные системы управления — предельный по сложности класс систем управления, ориентированных на получение, обработку и использование некоторой дополнительной информации, накапливаемой в базе знаний, предназначенные для работы в условиях неопределенности информации о свойствах сложных объектов и среды их функционирования и способные к адаптации и самообучению [11].

Адаптивность применительно к системам управления в первую очередь подразумевает интеллектуальные способности системы. Традиционные системы сетевого управления строятся как МАС, выполняющие простейшую работу по сбору данных с сетевых устройств с примитивной предварительной обработкой и передачей данных на мощный интеллектуальный центр (сервер управления).

Архитектура распределенной информационной многоагентной системы МКА

Ниже предлагается вариант решения задачи создания наземной и космической инфраструктуры управления и получения целевой информации в системах МКА на основе технологии МАС.

Разработка информационной интеллектуальной МАС включает в себя следующие элементы:

- создание модели взаимодействия интеллектуальных агентов;
- разработку методов визуальной спецификации знаний;
- разработку инструментальных средств распараллеливания логических программ и генерации интеллектуальных агентов;
- разработку средств координации взаимодействия интеллектуальных агентов.

Архитектура и функции распределенной информационной многоагентной системы (рис. 2), в зависимости от специфики обрабатываемой информации, включают несколько типов интеллектуальных агентов. Приведем основные из них:

1. Агент–координатор (менеджер проекта), согласует и связывает агентов между собой, выдает сведения о различных агентах (Coordinator Agent).

2. Агент доступа к базам данных (Data Base Agent).

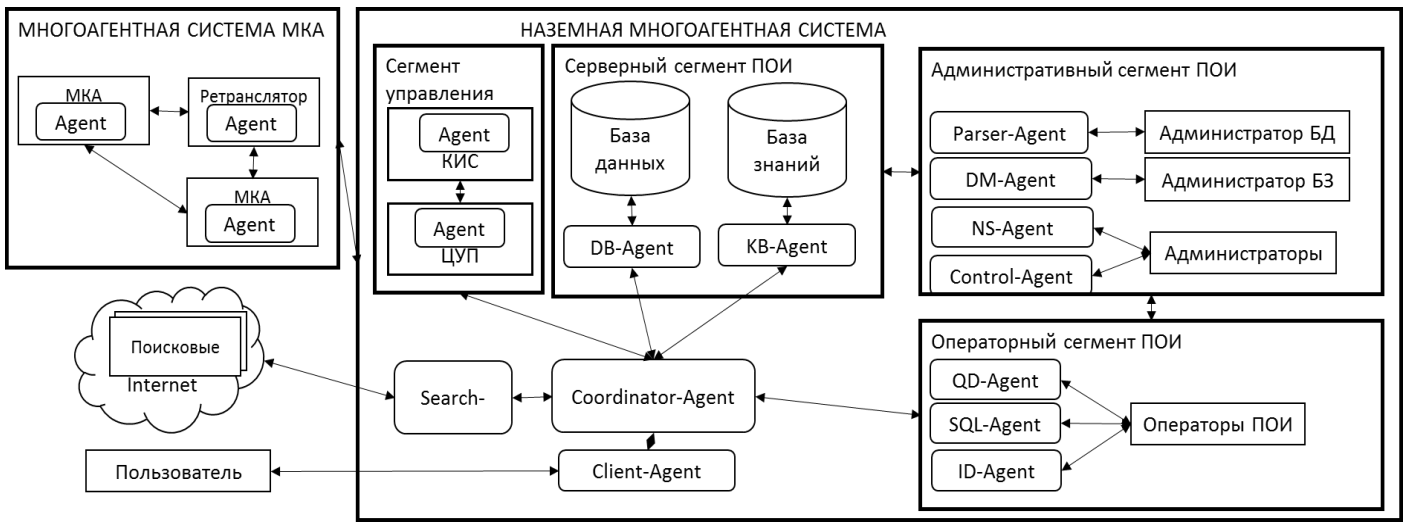


Рис. 2. Архитектура распределенной информационной многоагентной системы МКА

3. Агент доступа к базам знаний в предметной области, обеспечивает доступ к базе знаний (Knowledge Base Agent).

4. Поисковый агент, ищет необходимую информацию в Интернете, заносит ее в базу данных (Search Agent).

5. Агент тестирования и проработки информации в базе данных (Parser Agent).

6. Клиентский агент (агент пользователя), обеспечивает связь пользователей с базами данных/знаний (Client Agent).

7. Интерфейсный агент, помогающий клиенту ввести информацию в базу данных (Input Data Agent).

8. Агент составления запросов к базе данных (Query Data Agent)

9. Агент исполнения запросов к базе данных (SQL-Agent).

10. Агент управления сервером, или агент системного администратора (Control Agent).

11. Агент проверки непротиворечивости данных (Data Monitoring Agent).

12. Агент безопасности сети (Network Safety Agent).

В настоящее время существуют разные варианты программной реализации агентов: на C++, Java, HTML и т. д. Однако до сих пор нет единых стандартов для обеспечения совместной работы агентов. Пока существует только рабочий стандарт взаимодействия агентов ARPA Sharing Knowledge, основанный на языке ACL (Agent Communication Language).

Для управления системой МКА на основе многоагентной концепции существует возможность одновременного контроля за большим количеством меняющихся источников информации в автоматическом режиме. Ресурсы распределенных информационных систем на основе высокопроизводительной базы знаний будут доступны пользователям, обладающим программно-техническими средствами удаленного доступа через сеть Интернет, посредством интеллектуальных агентов, загружаемых с серверов пунктов обработки информации (ПОИ). В пределах ПОИ рекомендуется использование интеллектуальных агентов для поиска информации по intranet-технологии. В рамках распределенных интеллектуальных систем предлагается применять программные кооперативные МАС для сбора и обработки разнородной информации. При этом агент должен уметь осуществлять семантический поиск и сбор информации с учетом требований пользователя и состояния контролируемых объектов. В качестве средств доступа пользователей к услугам и информации, полученной с МКА, используется система интеллектуальных агентов, каждый из которых способен предоставлять пользователю собранную всеми агентами информацию, временно выполняя роль менеджера системы мониторинга. При этом агенты взаимодействуют на уровне знаний, а не данных. Агенты также способны перемещаться по узлам сети и менять свою численность в зависимости от разных условий.

Одной из важнейших проблем построения МАС МКА является проблема построения устойчивой коммуникационной системы [20]. Для этого необходимо обеспечить устойчивость самих каналов связи, а также взаимодействия между спутниками кластера в условиях возможного изменения пространственной конфигурации и условий прямой видимости. При этом необходимо предусмотреть возможность оперативной передачи потока задач между спутниками, как это реализовано в вычислительных кластерах балансировки нагрузки, например на основе концепции присоединения к кратчайшей очереди и перехода между очередями [25, 26]. Такая концепция адаптивной организации взаимодействия МКА предусматривает различные нештатные ситуации, в том числе перегрузку или ухудшение каналов связи, выход из строя (отказ) бортовой аппаратуры одного или нескольких спутников, изменение объема и характера решаемых задач и т. д.

Заключение

Таким образом, концепция МАС, вместе с существующими современными информационными технологиями, может быть использована для реализации инфраструктуры управления системой МКА. В настоящее время актуальной является задача поиска алгоритмов функционирования интеллектуальных агентов, обеспечивающих получение максимума информации при минимальных затратах ресурсов.

В работе сформулирована проблема построения самоорганизующейся системы группового управления поведением кластера МКА, реализующего полностью автономное исполнение заявок на сервис по добыванию информации о наземных объектах средствами космического наблюдения МКА. Предложена концепция построения, базовые принципы и компоненты программной архитектуры ее реализации. Реализация описанной концепции приведет к созданию космических систем МКА, которые будут способны автономно решать все задачи планирования группового поведения спутников кластера МКА и оперативного управления исполнением спланированной миссии с помощью бортовых вычислительных средств.

Литература

1. Концептуальные вопросы создания и применения малых космических аппаратов / Л. А. Макриденко, С. Н. Волков, В. П. Ходненко, С. А. Золотой // Вопросы электромеханики. Труды ВНИИЭМ. 2010. Т. 114, № 1. С. 15–26.

2. Гансвинд, И. Н. Малые космические аппараты — новое направление космической деятельности // Международный научно-исследовательский журнал. 2018. № 12-2 (78). С. 84–91. DOI: 10.23670/IRJ.2018.78.12.053.

3. Кокцинская, Е. М. Сверхмалые космические аппараты — наноспутники «CubeSat» // Видеонаука. 2018. № 1 (9). URL: http://videonauka.ru/stati/33-informatsionnye-soobshcheniya/183-sverkhmalye-kosmicheskie-apparaty-nanosputniki (дата обращения 01.06.2022).

4. Гончаренко, В. А. Обеспечение устойчивости функционирования наземного комплекса управления сверхмалыми космическими аппаратами // Микротехнологии и новые информационные услуги в авиации и космонавтике: Материалы IV Международной конференции (Санкт-Петербург, Россия, 06 октября 2005 г.). — Санкт-Петербург: Изд-во СПбГПУ, 2005. — С. 154–157.

5. Клюшников, В. Ю. Наноспутники — наиболее перспективный класс малых космических аппаратов / В. Ю. Клюшников, С. А. Клементьев // Инноватика и экспертиза. 2016. Вып. 2 (17). С. 97–105.

6. Басыров, А. Г. Методика функционально-распределенной обработки информации в перспективных кластерах микроспутников дистанционного зондирования Земли / А. Г. Басыров, В. В. Широбоков // Наукоемкие технологии в космических исследованиях Земли. 2017. Т. 9, № 4. С. 62–70.

7. Потюпкин, А. Ю. Кластеры малоразмерных космических аппаратов как новый тип космических объектов / А. Ю. Потюпкин, Н. С. Данилин, А. С. Селиванов // Ракетно-космическое приборостроение и информационные системы. 2017. Т. 4, Вып. 4. С. 45–56.
DOI: 10.17238/issn2409-0239.2017.4.45.

8. Широбоков, В. В. Подход к организации межспутникового взаимодействия в распределенной вычислительной структуре орбитальной группировки микроспутников / В. В. Широбоков, А. Ф. Шинкаренко // Труды Военно-космической академии им. А. Ф. Можайского. 2015. Вып. 646. С. 77–82.

9. Реализация автономного распределенного управления орбитальными системами космических аппаратов с использованием современных информационных технологий / А. А. Селиванов, А. В. Фарафонов, А. В. Ляшевский, Е. А. Миронов // Информация и космос. 2018. № 4. С. 154–158.

10. Гончаренко, В. А. Концептуальные основы построения устойчивых к воздействиям автоматизированных систем специального назначения на основе адаптивных технологий // Наукоемкие технологии в космических исследованиях Земли. 2018. Т. 10, № 4. С. 38–47.
DOI: 10.24411/2409-5419-2018-10095.

11. Гончаренко, В. А. Концептуальные основы комплексного интеллектуального управления распределенными информационно-вычислительными системами Космических войск // Военно-космическая деятельность России — истоки, состояние, перспективы: Сборник трудов научно-практической конференции (Россия, Санкт-Петербург, 30 марта–01 апреля 2005 г.). — Санкт-Петербург: ВКА имени А. Ф. Можайского, 2005. — Т. 2. — С. 146–148.

12. В РКС разрабатывают перспективную технологию управления многоспутниковой орбитальной группировкой // Aviation Explorer — Содружество авиационных экспертов. — 2020. — 18 ноября. URL: http://www.aex.ru/news/2020/11/18/219379/ (дата обращения 01.06.2022).

13. ЕТРИС ДЗЗ — современные решения в развитии отечественной наземной космической инфраструктуры дистанционного зондирования Земли из космоса / В. В. Ромашкин, П. А. Лошкарев, Д. И. Федоткин, [и др.] // Современные проблемы дистанционного зондирования Земли из космоса. 2019. Т. 16, № 3. С. 220–227.
DOI: 10.21046/2070-7401-2019-16-3-220-227.

14. Гончаренко, В. А. Применение многоагентных технологий построения наземной инфраструктуры систем малых космических аппаратов // Малые космические аппараты. Вып. 1. Принципы построения орбитальных систем и бортовых комплексов управления. — Санкт-Петербург: Министерство обороны РФ, 2001. — С. 25–28.

15. Клюшников, В. Ю. Построение кластеров малых космических аппаратов // Известия высших учебных заведений. Приборостроение. 2016. Т. 59, № 6. С. 423–428.
DOI: 10.17586/0021-3454-2016-59-6-423-428.

16. Монахова, У. В. Формирование роя наноспутников с помощью децентрализованного аэродинамического управления с учетом коммуникационных ограничений / У. В. Монахова, Д. С. Иванов; Институт прикладной математики имени М. В. Келдыша Российской академии наук. — Москва: ИПМ им. М. В. Келдыша РАН, 2018. — 32 с. — (Препринты ИПМ им. М. В. Келдыша; № 151).
DOI: 10.20948/prepr-2018-151.

17. Петренко, С. А. Проблема устойчивости функционирования киберсистем в условиях деструктивных воздействий // Труды Института системного анализа Российской академии наук. 2010. Т. 52. С. 68–105.

18. Гончаренко, В. А. Концептуальная модель обеспечения устойчивости функционирования распределенных информационных систем военного назначения на основе адаптивных технологий // Современные проблемы создания и эксплуатации вооружения, военной и специальной техники: Сборник статей III Всероссийской научно-практической конференции (Санкт-Петербург, Россия, 14–15 декабря 2016 г.). — Санкт-Петербург: ООО «Арт-Экспресс», 2016. — Т. 1. — С. 250–254.

19. Гончаренко, В. А. Оценивание устойчивости информационно-телекоммуникационных сетей к деструктивным воздействиям на основе учета связности маршрутов // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2015. № 1. С. 96–100.

20. Городецкий, В. И. Самоорганизация группового поведения кластера малых спутников распределенной системы наблюдения / В. И. Городецкий, О. В. Карсаев // Известия ЮФУ. Технические науки. 2017. № 2 (187). С. 234–247. DOI: 10.18522/2311-3103-2017-1-234247.

21. Wooldridge, M. J. Agent Theories, Architectures, and Languages: A survey / M. J. Wooldridge, N. R. Jennings // Proceedings of ECAI-94 Workshop on Agent Theories, Architectures, and Languages (ATAL 1994) (Amsterdam, Netherlands, 08–09 August 1994) / M. J. Wooldridge, N. R. Jennings (eds.) // Lecture Notes in Artificial Intelligence. Lecture Notes in Computer Science. Vol. 890. — Berlin: Springer-Verlag, 1995. — Pp. 1–39. DOI: 10.1007/3-540-58855-8_1.

22. Манько, С. В. Концепция построения мультиагентных робототехнических систем / С. В. Манько, В. М. Лохин, М. П. Романов // Вестник МГТУ МИРЭА. 2015. № 3-1(8). С. 156–165.

23. Городецкий, В. И. Управление коллективным поведением роботов в автономной миссии // Робототехника и техническая кибернетика. 2016. № 1 (10). С. 40–54.

24. Калюжный, А. В. Проблемы повышения живучести робототехнических систем космического назначения на основе многоагентных технологий / А. В. Калюжный, В. А. Гончаренко, А. Н. Соколовский // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2020. № 1. С. 168–173.
DOI: 10.25586/RNU.V9187.20.01.P.168.

25. Гончаренко, В. А. Модели адаптивного перераспределения нагрузки в кластерных вычислительных системах // Известия высших учебных заведений. Приборостроение. 2008. Т. 51, № 3. С. 32–37.

26. Авторское свидетельство № 1509928 СССР, G 06 F 15/20. Устройство для моделирования двухканальной системы массового обслуживания: № 4364699/24-24: заявл. 13.01.1988: опубл. 23.09.1989 / Гончаренко В. А., Филимонихин Г. В.; заявитель Военный инженерный краснознаменный институт имени А. Ф. Можайского. — 6 с.

DOI: 10.24412/2413-2527-2022-230-32-40

Methods for Constructing Cluster Space Systems Based on Multi-Agent Technologies

PhD V. A. Goncharenko
Emperor Alexander I St. Petersburg
State Transport University,
Mozhaisky Military Space Academy
Saint Petersburg, Russia
vlango@mail.ru

PhD V. S. Zabuzov
Mozhaisky Military Space Academy
Saint Petersburg, Russia
zzvvss80@mail.ru

Abstract. **The basic principles and methods of constructing a system of small spacecraft in the form of a cluster based on multi-agent technologies are proposed. The main approaches to improving the stability of the functioning of small spacecraft systems are considered. Architectural solutions for the construction of a multi-agent system of small spacecraft are proposed, which is based on the concept of clustering of onboard computing system and multi-functional typology of agents.**

Keywords: **multi-agent system, satellite cluster, small spacecraft, intelligent agent, stability of functioning, survivability, distributes system, on-board computer system.**

REFERENCES

1. Makridenko L. A., Volkov S. N., Khodnenko V. P., Zolotoy S. A. Conceptual Problems on Creation and Application of Small Spacecraft [Kontseptualnye voprosy sozdaniya i primeneniya malykh kosmicheskikh apparatov], *Electromechanical Matters. VNIIEM Studies [Voprosy elektromekhaniki. Trudy VNIIEM],* 2010, Vol. 114, No. 1, Pp. 15–26.

2. Ganswind I. N. Small Spacecraft — New Direction in Space Activities [Malye kosmicheskie apparaty — novoe napravlenie kosmicheskoy deyatelnosti], *International Research Journal [Mezhdunarodnyy nauchno-issledovatelskiy zhurnal]*, 2018, No. 12-2 (78), Pp. 84–91.
DOI: 10.23670/IRJ.2018.78.12.053.

3. Koktsinskaya Ye. M. Ultra-Small Space Vehicles — Nano-Satellites «CubeSat» [Sverkhmalye kosmicheskie apparaty — nanosputniki «CubeSat»], *Video Science [Videonauka]*, 2018, No. 1 (9). Available at: http://videonauka.ru/stati/33-informatsionnye-soobshcheniya/183-sverkhmalye-kosmicheskie-apparaty-nanosputniki (accessed 01 Jun 2022).

4. Goncharenko V. A. Ensuring the Stability of the Functioning of the Ground-Based Control Complex for Ultra-Small Spacecraft [Obespechenie ustoychivosti funktsionirovaniya nazemnogo kompleksa upravleniya sverkhmalymi kosmicheskimi apparatami], *Microtechnologies and New Information Services in Aviation and Cosmonautics: Proceedings of the IV International Conference [Mikrotekhnologii i novye informatsionnye uslugi v aviatsii i kosmonavtike: Materialy IV Mezhdunarodnoy konferentsii], Saint Petersburg, Russia, October 06, 2005*. St. Petersburg, Peter the Great St. Petersburg Polytechnic University, 2005, Pp. 154–157.

5. Klyushnikov V. Yu., Klementyev S. A. Nanosatellites — The Most Promising Small Spacecraft [Nanosputniki — naibolee perspektivnyy klass malykh kosmicheskikh apparatov], *Innovatics and Expert Examination [Innovatika i ekspertiza],* 2016, Vol. 2 (17), Pp. 97–105.

6. Basyrov A. G., Shirobokov V. V. Methodology of Functionally-Distributed Information Processing in the Prospective Clusters of Microsatures of Remote Sensing of the Earth [Metodika funktsionalno-raspredelennoy obrabotki informatsii v perspektivnykh klasterakh mikrosputnikov distantsionnogo zondirovaniya Zemli], *High Tech in Earth Space Research [Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli]*, 2017, Vol. 9, No. 4. Pp. 62–70.

7. Potyupkin A. Yu., Danilin N. S., Selivanov A. S. Small Satellites Clusters — A New Type of Space Objects [Klastery malorazmernykh kosmicheskikh apparatov kak novyy tip kosmicheskikh obektov], *Rocket-Space Device Engineering and Information Systems [Raketno-kosmicheskoe priborostroenie i informatsionnye sistemy],* 2017, Vol. 4, Is. 4, Pp. 45–56.
DOI: 10.17238/issn2409-0239.2017.4.45.

8. Shirobokov V. V., Shinkarenko A. F. Approach to Organizing Inter-Satellite Interaction in the Distributed Computing Structures Orbital Constellation of Microsatellites [Podkhod k organizatsii mezhsputnikovogo vzaimodeystviya v raspredelennoy vychislitelnoy strukture orbitalnoy gruppirovki mikrosputnikov], *Proceedings of the Mozhaisky Military Space Academy [Trudy Voenno-kosmicheskoy akademii imeni A. F. Mozhayskogo],* 2015, Is. 646, Pp. 77–82.

9. Selivanov A. A., Farafonov A. V., Lyashevsky A. V., Mironov E. A. Design of Autonomous Distributed Control for Spacecraft Orbital Systems Based on Advanced IT Solutions [Realizatsiya avtonomnogo raspredelennogo upravleniya orbitalnymi sistemami kosmicheskikh apparatov s ispolzovaniem sovremennykh informatsionnykh tekhnologiy], *Information and Space [Informatsiya i kosmos]*, 2018, No. 4, Pp. 154–158.

10. Goncharenko V. A. Conceptual Bases for Building Resilient to the Impacts of Automated Systems for Special Purposes Based on Adaptive Technologies [Kontseptualnye osnovy postroeniya ustoychivykh k vozdeystviyam avtomatizirovannykh sistem spetsialnogo naznacheniya na osnove adaptivnykh tekhnologiy], *High Tech in Earth Space Research [Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli]*, 2018, Vol. 10, No. 4, Pp. 38–47.
DOI: 10.24411/2409-5419-2018-10095.

11. Goncharenko V. A. Conceptual Foundations of Integrated Intelligent Control of Distributed Information and Computing Systems of Space Troops [Kontseptualnye osnovy kompleksnogo intellektualnogo upravleniya raspredelennymi informatsionno-vychislitelnymi sistemami Kosmicheskikh voysk], *Military Space Activity of Russia — Origins, State, Prospects: Proceedings of a Scientific and Practical Conference [Voenno-*

kosmicheskaya deyatelnost Rossii — istoki, sostoyanie, perspektivy: Sbornik trudov nauchno-prakticheskoy konferentsii], Saint Petersburg, Russia, March 30–April 01, 2005. St. Petersburg, Mozhaisky Military Space Academy, 2005, Vol. 2, Pp. 146–148.

12. The RSS is Developing a Promising Technology for Controlling a Multi-Satellite Orbital Grouping [V RKS razrabatyvayut perspektivnuyu tekhnologiyu upravleniya mnogosputnikovoy orbitalnoy gruppirovkoy], *Aviation Explorer — Commonwealth of Aviation Experts [Aviation Explorer — Sodruzhestvo aviatsionnykh ekspertov]*. Published online at November 18, 2020. Available at: http://www.aex.ru/news/2020/11/18/219379/ (accessed 01 Jun 2022).

13. Romashkin V. V., Loshkarev P. A., Fedotkin D. I., et al. UGDIS ERS — Modern Solutions in the Development of Domestic Terrestrial Space Infrastructure of Earth Remote Sensing from Space [ETRIS DZZ — sovremennye resheniya v razvitii otechestvennoy nazemnoy kosmicheskoy infrastruktury distantsionnogo zondirovaniya Zemli iz kosmosa], *Current Problems in Remote Sensing of the Earth from Space [Sovremennye problemy distantsionnogo zondirovaniya Zemli iz kosmosa]*, 2019, Vol. 16, No. 3, Pp. 220–227.
DOI: 10.21046/2070-7401-2019-16-3-220-227.

14. Goncharenko V. A. Application of Multi-Agent Technologies for Building Ground Infrastructure for Small Spacecraft Systems [Primenenie mnogoagentnykh tekhnologiy postroeniya nazemnoy infrastruktury sistem malykh kosmicheskikh apparatov], *Small Spacecraft. Issue 1. Principles of Construction of Orbital Systems and Onboard Control Systems [Malye kosmicheskie apparaty. Vypusk 1. Printsipy postroeniya orbitalnykh sistem i bortovykh kompleksov upravleniya].* Saint Petersburg, Ministry of Defense of the Russian Federation, 2001, Pp. 25–28.

15. Klyushnikov V. Yu. Construction of Small Spacecraft Clusters [Postroenie klasterov malykh kosmicheskikh apparatov], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 2016, Vol. 59, No. 6, Pp. 423–428. DOI: 10.17586/0021-3454-2016-59-6-423-428.

16. Monakhova U. V., Ivanov D. S. Swarm of Nanosatellites Deployment by Decentralized Aerodynamic Drag Control with Communicational Constraints [Formirovanie roya nanosputnikov s pomoshchyu detsentralizovannogo aerodinamicheskogo upravleniya s uchetom kommunikatsionnykh ogranicheniy], *KIAM Preprint [Preprinty IPM imeni M. V. Keldysha]*, No. 151. Moscow, Keldysh Institute of Applied Mathematics of RAS, 2018, 32 p. DOI: 10.20948/prepr-2018-151.

17. Petrenko S. A. The Problem of Stability of Functioning of Cybersystems in Conditions of Destructive Actions [Problema ustoychivosti funktsionirovaniya kibersistem v usloviyakh destruktivnykh vozdeystviy], *Proceeding of the Institute for Systems Analysis of the Russian Academy of Science [Trudy Instituta sistemnogo analiza Rossiyskoy akademii nauk]*, 2010, Vol. 52. Pp. 68–105.

18. Goncharenko V.A. Conceptual Model of Ensuring the Stability of the Functioning of Distributed Military Information Systems Based on Adaptive Technologies [Kontseptualnaya model obespecheniya ustoychivosti funktsionirovaniya raspredelennykh informatsionnykh sistem voennogo naznacheniya na osnove adaptivnykh tekhnologiy], *Modern Problems of Creation and Operation of Weapons, Military and Special Equipment: Proceedings of the III All-Russian Scientific and Practical Conference [Sovremennye problemy sozdaniya i ekspluatatsii vooruzheniya, voennoy i spetsialnoy tekhniki: Sbornik statey III Vserossiyskoy nauchno-prakticheskoy konferentsii], Saint Petersburg, Russia, December 14–15, 2016*, *Volume 1*. Saint Petersburg, Art-Express, 2016, Pp. 250–254.

19. Goncharenko V.A. Evaluation of Information Telecommunication Networks Stability to Destructive Effects Based on the Connectivity of Routes [Otsenivanie ustoychivosti informatsionno-telekommunikatsionnykh setey k destruktivnym vozdeystviyam na osnove ucheta svyaznosti marshrutov], *Vestnik of Russian New University. Series «Complex Systems: Models, Analysis, Management» [Vestnik Rossiyskogo novogo universiteta. Seriya «Slozhnye sistemy: modeli, analiz i upravlenie»]*, 2015, No. 1, Pp. 96–100.

20. Gorodetsky V. I., Karsaev O. V. Distributed Surveillance System Based on Self-Organized Collective Behavior of Small Satellite Cluster [Samoorganizatsiya gruppovogo povedeniya klastera malykh sputnikov raspredelennoy sistemy nablyudeniya], *Izvestiya SFedU. Engineering Sciences [Izvestiya YuFU. Tekhnicheskie nauki]*, 2017, No. 2 (187), Pp. 234–247. DOI: 10.18522/2311-3103-2017-1-234247.

21. Wooldridge M. J., Jennings N. R. Agent Theories, Architectures, and Languages: A survey. *In: Wooldridge M. J., Jennings N. R. (eds.) Proceedings of ECAI-94 Workshop on Agent Theories, Architectures, and Languages (ATAL 1994), Amsterdam, Netherlands, 08–09 August 1994. Lecture Notes in Artificial Intelligence. Lecture Notes in Computer Science*. Vol. 890. Berlin, Springer-Verlag, 2005. Pp. 1–39.
DOI: 10.1007/3-540-58855-8_1.

22. Manko S. V., Lokhin V. M., Romanov M. P. The Concept of Multi-Agent Robotic Systems [Kontseptsiya postroeniya multiagentnykh robototekhnicheskikh sistem], *MSTU MIREA Herald [Vestnik MGTU MIREA]*, 2015, No. 3-1 (8), Pp. 156–165.

23. Gorodetsky V. I. Robot Cooperative Behaviour Control During Autonomous Mission [Upravlenie kollektivnym povedeniem robotov v avtonomnoy missii], *Robotics and Technical Cybernetics [Robototekhnika i tekhnicheskaya kibernetika]*, 2016, No. 1 (10), Pp. 40–54.

24. Kalyuzhnyj A. V., Goncharenko V. A., Sokolovskij A. N. The Problem of Increasing the Survivability of Robotic Systems for Space Applications Based on Multi-Agent Technologies [Problemy povysheniya zhivuchesti robototekhnicheskikh sistem kosmicheskogo naznacheniya na osnove mnogoagentnykh tekhnologiy], *Vestnik of Russian New University. Series «Complex Systems: Models, Analysis, Management» [Vestnik Rossiyskogo novogo universiteta. Seriya «Slozhnye sistemy: modeli, analiz i upravlenie»]*, 2020. No. 1. Pp. 168–173.
DOI: 10.25586/RNU.V9187.20.01.P.168.

25. Goncharenko V. A. Models of Adaptive Load Redistribution in Cluster Computing Systems [Modeli adaptivnogo pereraspredeleniya nagruzki v klasternykh vychislitelnykh sistemakh], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 2008, Vol. 51, No. 3, Pp. 32–37.

26. Goncharenko V. A., Filimonikhin G. V. Device for Modeling a Two-Channel Queuing System [Ustroystvo dlya modelirovaniya dvukhkanalnoy sistemy massovogo obsluzhivaniya], Certificate of Authorship SU No. 1509928, published at September 23, 1989, 6 p.

DOI: 10.24412/2413-2527-2022-230-41-47

Применение нейронной сети YOLO для распознавания дефектов

Э. А. Брехт, к.т.н. В. Н. Коншина
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
ed.breht@mail.ru

Аннотация. **В статье рассматривается пример использования нейронной сети YOLO для распознавания и визуализации дефектов на изображении, полученном в процессе ультразвукового контроля дифракционно-временным методом. Из 110 дефектограмм, содержащих изображения дефектов, 100 использовались для обучения нейронной сети, 10 — для проверки. Показана возможность распознавания дефектов нейронной сетью с достаточной точностью для повседневной практики неразрушающего контроля.**

Ключевые слова: **нейронная сеть, ультразвуковой контроль, дефектограммы, распознавание.**

Введение

В ультразвуковом контроле больше всего возможностей для реализации методов искусственного интеллекта. В первую очередь это связано с тем, что при ультразвуковом контроле информация с первичных преобразователей поступает в виде сигналов, которые можно непосредственно обрабатывать, а не в виде визуальных изображений, как, например, в радиографии. Кроме того, ультразвуковой контроль реализуется различными методами (метод отраженного излучения (эхо-метод), метод прошедшего излучения и т. д.) [1], для каждого из которого характерны свои измеряемые характеристики дефектов [2], а также варианты методов, отличающиеся значениями основных параметров. Многообразие вариантов метода способствует появлению новых задач, связанных с представлением и обработкой информации о процессе контроля. Кроме того, существуют и направления классификации методов по назначению: обнаружение дефектов, выявление коррозии, определение структуры металла, оценка качества сварных швов, измерения толщины и т. д.

На сегодняшний день известно о следующих задачах применительно к ультразвуковому контролю, в рамках которых были предприняты попытки использовать нейронные сети [3]: распознавание образов, разделение сигналов от дефектов различного типа, автоматизированный анализ дефектограмм.

Для решения приведенных выше задач использовались уникальные архитектурно-программные решение (программы для работы с технологией искусственного интеллекта), что в свою очередь и является недостатком, так как в настоящее время большинство упомянутых в [3] решений не используется, поскольку современные решения [4] имеют сравнительно более высокий уровень адаптивности (гибкости). Основной особенностью и преимуществом нейронных сетей перед традиционными алгоритмами — это возможность обучения. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. То есть, при успешном обучении, сеть позволит получить результат на основании данных, которые отсутствовали в обучающей выборке, а также в случае представления неполных и/или «зашумленных», частично искаженных данных.

Архитектура нейронной сети YOLO является примером современного гибкого решения задач по распознаванию и анализу дефектов. Стоит также отметить, что в сравнении с упомянутыми в [3] технологиями нейронная сеть YOLO продолжает развиваться и является системой с открытым исходным кодом, то есть бесплатной.

Постановка задачи по распознаванию дефектов

Задачи по распознаванию дефектов являются наиболее популярными и актуальными в рамках ультразвукового контроля, позволяя повысить достоверность его результатов за счет снижения субъективности. В развертке типа В (зависимость амплитуды от времени прихода сигнала) оценить наличие дефекта можно по геометрическим и амплитудным признакам, что в свою очередь дает возможность для применения нейронных сетей. В рамках данной работы были использованы результаты контроля дифракционно-временным методом.

Задачей работы являлась разработка программы, способной определять и визуализировать наличие дефекта на дефектограмме с развёрткой типа В, полученной при контроле дифракционно-временным методом.

Дефектограмма, полученная в процессе проведения контроля, приведена на рисунке 1.

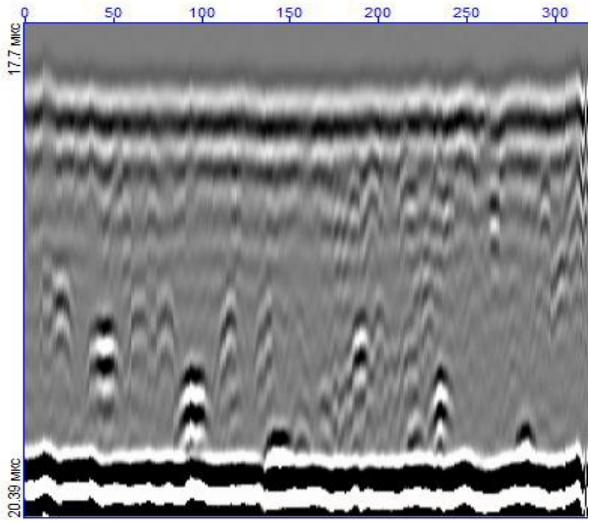


Рис. 1. Дефектограмма, полученная в результате контроля дифракционно-временным методом

Как указано выше, необходимо оценить наличие дефекта по геометрическим признакам. Таким образом необходимо выделить геометрическую форму, по которой можно оценить наличие дефекта (рис. 2).

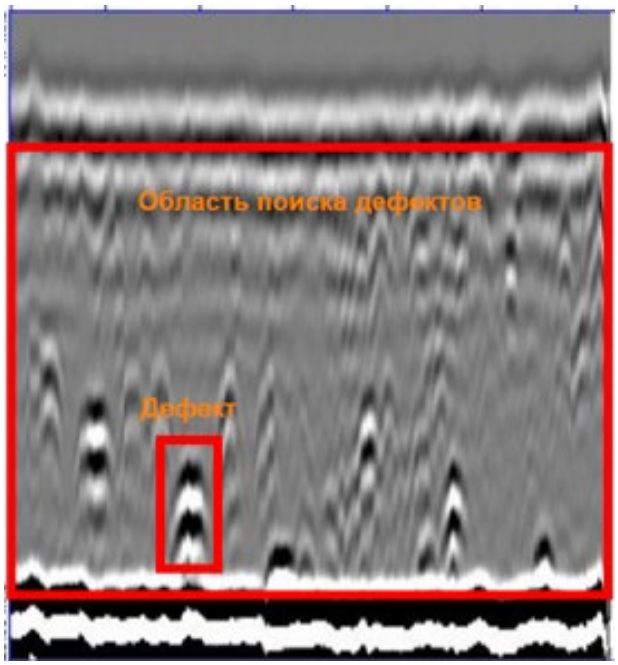


Рис. 2. Пример геометрической формы распознаваемого дефекта

На изображении отмечены не все видимые дефекты, однако в данном случае отмеченный дефект иллюстрирует постановку задачи, а не конечный результат.

Особенности работы нейронной сети YOLO

Существует немало архитектур нейронных сетей, позволяющих распознавать сущности (дефекты) на изображении [5]. Большинство из них представляют собой двухуровневые нейронные сети, когда одна нейронная сеть сочетает в себе несколько более простых. Основной принцип работы таких нейронных сетей заключается в разделении всего изображения на регионы или клетки, после чего происходит анализ каждого региона.

У двухуровневых нейронных сетей есть две существенные проблемы: изображение анализируется не целиком, а только выделенные регионы, а скорость работы таких нейронных сетей значительно ниже. Первый недостаток связан с принципом работы, упомянутым ранее. Зачастую каждый регион обрабатывается последовательно с помощью сверточной нейронной сети. Результат, полученный сверточной нейронной сетью, передается в полносвязную для определения класса и размеров области, внутри которой находится искомый объект. Далее нейронная сеть обрабатывает следующий регион, и алгоритм повторяется.

Однако существует нейронная сеть, которая лишена этих недостатков. Выбранная нейронная сеть YOLO (You Only Look Once) [6] представляет собой двухуровневую нейронную сеть, преимущественно со сверточными слоями. Забегая вперед, можно отметить, что YOLO способна обрабатывать все регионы, из которых состоит изображение, за один раз. Главная причина, по которой была выбрана именно эта нейронная сеть, заключается в ее эффективности, так как она имеет лучшие значения по соотношению между скоростью и точностью [6]. На рисунке 3 графически представлены возможности четвертой версии нейронной сети YOLO.

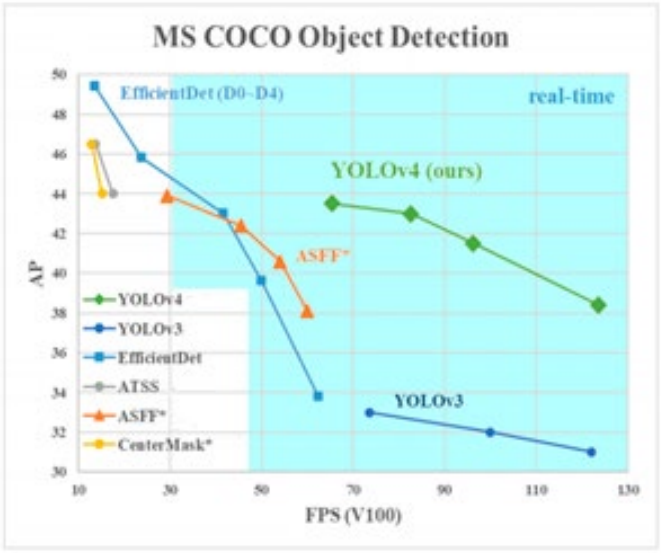


Рис. 3. Сравнение эффективности нейронных сетей

По оси абсцисс на рисунке 3 отложена скорость работы нейронной сети (частота кадров в секунду), по оси ординат — точность (измеряется с помощью специального коэффициента для измерения точности детекторов объекта [7]). Как видно из графика, четвертая версия YOLO существенно превосходит аналоги и показывает более стабильные значения.

Чтобы понять, каким образом нейронная сеть смогла достичь столь высоких показателей в соотношении «скорость — точность», необходимо рассмотреть уникальные особенности, которыми обладает данная нейронная сеть.

Функционирование нейронной сети YOLO основывается на работе с представленными на рисунке 4 клетками. Каждая клетка является «якорем», к которому прикрепляется рамка, в которой находится выделенный объект.

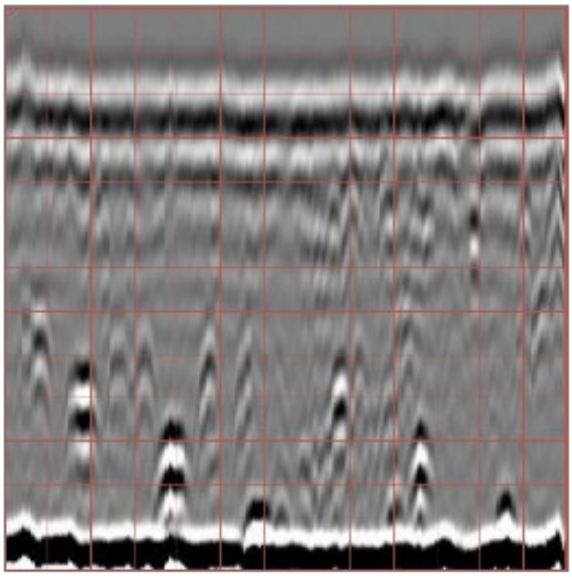


Рис. 4. Разделение исходного изображения на клетки

Для разделения загружаемого изображения на равные части необходимо, чтобы размер был кратным 32. Рекомендованным значением является 416 пикселей в длину и ширину. При большем разрешении увеличивается время обучения нейронной сети, однако это также положительно влияет на качество работы нейронной сети. Помимо этого, соблюдение данного требования позволяет загружать партию изображений в нейронную сеть за один раз, после чего применяется технология параллельных вычислений.

Распознав искомый участок на изображении, нейронная сеть рисует вокруг «якорной» клетки, в пределах которой находится искомый участок, несколько прямоугольников разного размера (рис. 5).

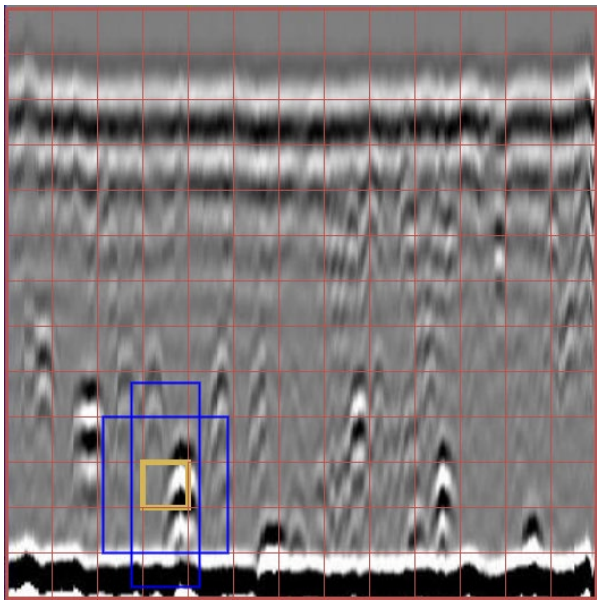


Рис. 5. Пример отрисовки прямоугольников

В первую очередь стоит определиться с размерами прямоугольников, внутри которых может находиться искомый объект. Максимальное и минимальное значения прямоугольников можно настроить в параметрах нейронной сети, указав конкретные значения, или позволив нейронной сети выбрать размеры из обучающей выборки. Как правило, применяют второй вариант.

Координаты рамок определяются несколько сложней. Вначале нейронная сеть рассчитывает центр искомой области внутри «якорной» клетки, далее относительно этого центра определяются координаты рамки (рис. 6). Для расчета нейронная сеть использует координату левой верхней «якорной» клетки, на которую попадает рамка.

Как было упомянуто ранее, нейронная сеть рисует несколько рамок вокруг искомого объекта, это необходимо для более точного выделения области, в которой находится объект. В результате проведения всех операций у одной выделенной области может быть несколько подходящих для финального отображения рамок (рис. 7).

Для оптимизации рамок с целью оставить только одну рамку для одного объекта (рис. 8) используется алгоритм обработки NMS (Non-Maximum Suppression), описанный в [8].

Если сравнить нейронную сеть YOLO с другими примерами нейронных сетей, то главной особенностью является принцип разделения всего изображения на секции, которыми являются «якорные» клетки.

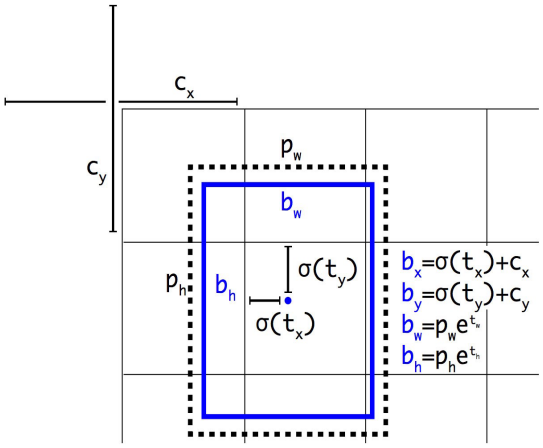


Рис. 6. Выделение распознаваемых областей

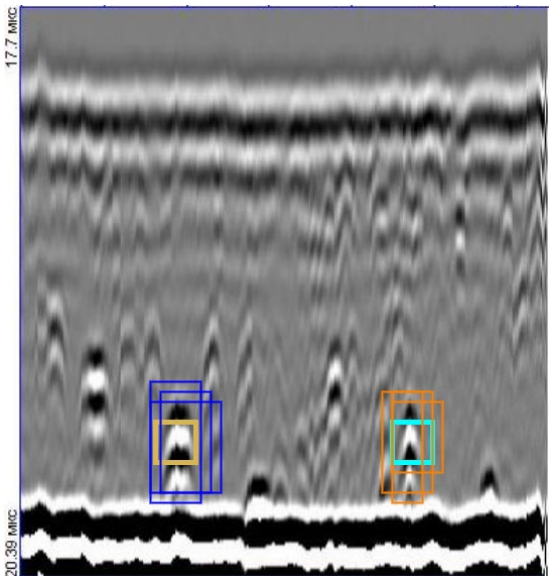


Рис. 7. Пример изображения с несколькими рамками распознанных объектов

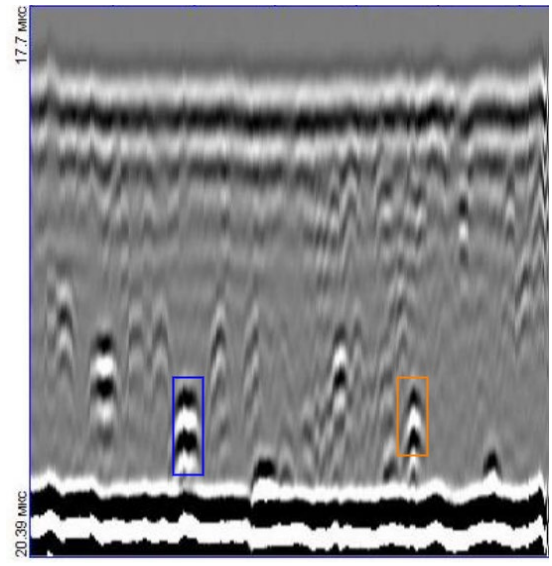


Рис. 8. Рамки распознанных объектов после применения алгоритма NMS

Если сравнить нейронную сеть YOLO с другими примерами нейронных сетей, то главной особенностью является принцип разделения всего изображения на секции, которыми являются «якорные» клетки.

Нейронная сеть EfficientDet [9] обладает хорошим показателем точности. Это достигается с помощью ее уникальной структуры, в основе которой использован принцип одноуровневой нейронной сети. Признаки, по которым нейронная сеть распознает объекты, передаются сразу по нескольким направлениям (в других нейронных сетях эта реализация более линейна). Очевидно, что для таких операций потребуется больше переменных, а также уникальные правила для обработки признаков.

Нейронная сеть ATSS (Adaptive Training Sample Selection) [10] является прямым противопоставлением нейронной сети YOLO, поскольку изначально создавалась как нейронная сеть для распознавания, которая не использует принцип «якорных» клеток. Вместо этого данная нейронная сеть использует точки. После того как искомый объект отмечен точками, производятся математические вычисления, определяющие размер рамки, внутри которой находится объект. Эта нейронная сеть может обрабатывать только одно изображение за один раз, что и является главным недостатком.

Нейронная сеть ASFF (Adaptively Spatial Feature Fusion) [11] является аналогом нейронной сети YOLO. Основные принципы работы схожи. Главное отличие заключается в математическом принципе, по которому нейронная сеть обрабатывает признаки для последующего распознавания объектов. В случае с YOLO обработка происходит по жестко заданным алгоритмам, в ASFF это происходит адаптивно.

Нейронная сеть CenterMask [12] с точки зрения технологий является предшественником YOLO. Основные принципы схожи с работой нейронной сети YOLO. Однако CenterMask более проста в исполнении и имеет меньший программный функционал. В первую очередь нейронная сеть обладает сравнительно меньшими сверточными слоями, что вынуждает использовать более сложные схемы для обработки признаков. В связи с упомянутыми факторами в нейронной сети CenterMask отсутствует возможность обрабатывать несколько изображений за один раз.

Рассмотрев все ранее упомянутые архитектуры нейронных сетей, можно выделить общую закономерность: скорость работы нейронной сети является не только явным преимуществом, но и образующим фактором, который определяет структуру и принцип работы нейронной сети.

Применение нейронной сети YOLO

Весь процесс работы нейронной сети YOLO можно разделить на следующие этапы:

1. Изображение подается на вход нейронной сети.

2. Изображение анализируется сверточной нейронной сетью, на выходе которой формируются карты признаков — значения, по которым нейронная сеть сопоставляет искомые области с участками, из которых состоит входное изображение.

3. Определяются места (регионы), в которых с высокой вероятностью находятся объекты, которые необходимо распознать.

4. Классифицируются участки, внутри которых находятся распознанные объекты.

Неотъемлемой частью работы нейронной сети является процесс обучения. Для того чтобы обучить нейронную сеть, необходимо иметь обучающую выборку. В данном случае использовалось 110 изображений, 100 из которых были выделены для обучения и 10 — для проверки точности. Важно отметить, что нейронные сети работают только с числами, поэтому на входе и выходе нейронной сети указываются соответствующие числа. В качестве числовых значений на вход подается матрица чисел, где каждому элементу матрицы соответствует значение яркости пикселя. В случае если изображение цветное, на вход подается три матрицы, по одной на каждый из основных цветов (красный, зеленый, синий). На выходе нейронная сеть возвращает массив значений, состоящий из координат области, в которой находится распознанный объект. Если объектов несколько, то на выходе — матрица значений, в которой количество рядов соответствует количеству распознанных объектов.

Выходные значения для обучающей выборки необходимо записать вручную, поскольку от точности выходных значений обучающей выборки напрямую зависит точность работы нейронной сети. Это легко сделать с помощью программы, позволяющей визуально выделить области на изображении и получить координаты выделенных областей (рис. 9).

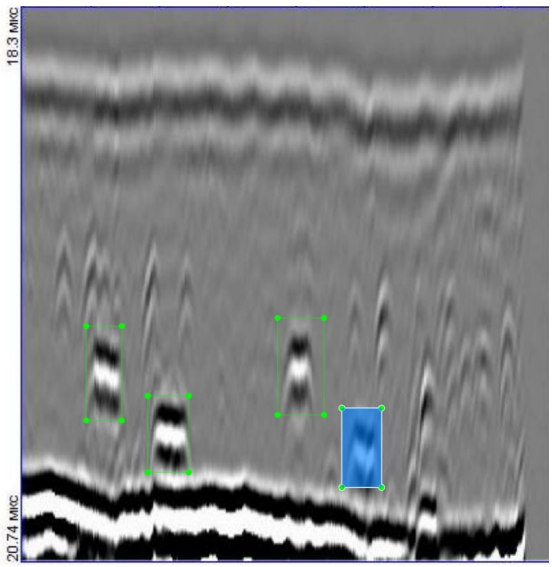


Рис. 9. Выделение распознаваемых областей

Технология параллельных вычислений, которую поддерживает YOLO, позволяет очень быстро провести процесс обучения.

Процесс обучения нейронной сети можно отслеживать с помощью графика, который представлен на рисунке 10. По оси абсцисс отмечено количество итераций, а по оси ординат — значение ошибки (разница между истинным выходным значением и значением, которое рассчитала нейронная сеть).

Чтобы не столкнуться с эффектом переобучения, нейронная сеть автоматически сохраняет (обновляет) значения параметров, при которых нейронная сеть имеет минимальное значение ошибки.

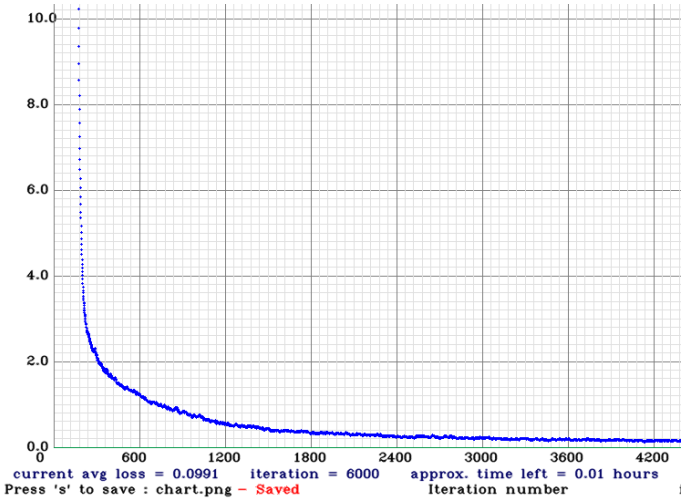


Рис. 10. График обучения нейронной сети

Как видно из графика, значение ошибки составляет 0,0991, что примерно соответствует 10 %. Однако стоит учитывать тот факт, что данный показатель основывается на данных, представленных для обучения, иными словами, параметры нейронной сети имеют такие значения, которые позволяют распознавать объекты на изображениях, представленных для обучения с вероятностью ошибки в 10 %. В связи с этим после обучения нейронной сети необходимо проверить эффективность работы на проверочной выборке, которая в данном случае состоит из 10 изображений.

Так как нейронная сеть на входе и выходе имеет числовые значения, то для визуализации полученного результата необходимо использовать программу, способную работать с этими числами. В данном примере была написана программа на языке Python, которая позволила задействовать конфигурацию обученной нейронной сети. Примеры использования программы и нейронной сети приведены на рисунках 11 и 12. Изображения различного качества были взяты намеренно, чтобы еще раз убедиться в эффективности работы нейронной сети.

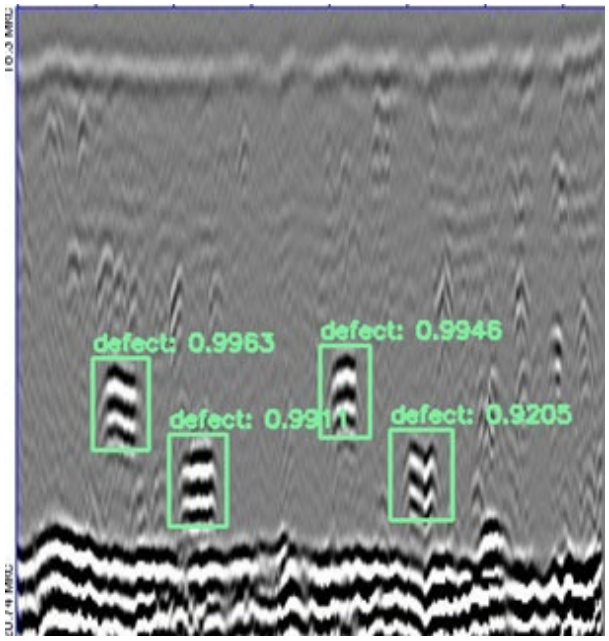


Рис. 11. Первый пример результата распознавания дефектов

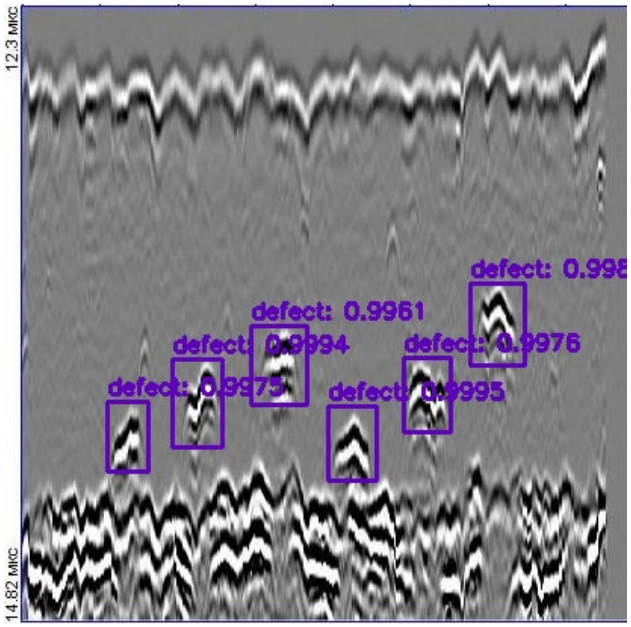


Рис. 12. Второй пример результата распознавания дефектов

Ранее было отмечено, что для обучения нейронной сети необходимо использовать изображения определенного размера. Однако когда нейронная сеть обучена, использовать изображения определенного размера необязательно. При этом стоит помнить, что размер изображения будет влиять на скорость распознавания.

Одним из требований к исходным данным для обучения нейросети является их однозначность и непротиворечивость, чтобы нейросеть не выдавала высокие вероятности отнесения одного объекта к нескольким классам.

Распознанные области выделяются рамкой. Текст, указанный над рамкой, является названием класса, к которому относится распознанный объект. Расположенные рядом с текстом цифры характеризуют точность. Как можно заметить, вероятность распознавания превышает 90 %. Аналогичные результаты распознавания получены и на оставшихся восьми изображениях. Анализируя результаты, стоит упомянуть о возможности верифицировать распознанный объект, задавая минимальное значение вероятности распознавания. Если указать минимальное значение вероятности распознавания, то все распознанные объекты, вероятность распознавания которых будет меньше заданного минимума, не будут отображаться на итоговом изображении.

Применение данного программного решения не ограничивается распознаванием и классификацией объектов на изображении. Спектр задач, которые возможно решить с помощью данной нейронной сети, напрямую зависит от программы, которая будет с ней взаимодействовать. Поскольку имеется возможность получить координаты рамок, возможно использовать данную информацию для последующих задач. В качестве примера можно отметить возможность представления информации в более понятном для человека виде, например, в какой части контролируемого объекта находится дефект.

Использование нейронной сети YOLO в режиме реального времени

Как отмечено ранее, нейронная сеть YOLO обладает преимуществом в соотношении скорости и точности. Результаты, представленные на рисунке 2, позволяют утверждать, что нейронная сеть способна работать в режиме реального времени. Кроме того, с точки зрения архитектуры в данном программном решении заложены специальные математические концепции, позволяющие отслеживать перемещение объекта на видео.

Для нейронной сети стоит задача не просто определить объекты на кадре, но еще и связать информацию с предыдущих кадров таким образом, чтобы не потерять объект или сделать его уникальным. Первый этап задачи сводится к выполнению рассмотренного ранее алгоритма. Однако ситуация сильно осложняется в случае необходимости передачи информации от кадра к кадру. Для решения данной задачи в архитектуру нейронной сети были заложены два математических аспекта — расстояние Махаланобиса [13] и фильтр Калмана [14].

Используя математические вычисления, нейронная сеть добавляет дополнительные параметры к распознанным на первом кадре объектам. Добавляется новая метрика, позволяющая последовательно идентифицировать объекты на каждом кадре.

Чтобы использовать нейронную сеть YOLO в режиме реального времени, требуется значительно большее количество вычислительных ресурсов, чем для распознавания объектов на изображении, что усложняет применение данной технологии в режиме реального времени.

Заключение

Нейронной сети YOLO удалось распознать дефекты на изображениях с точностью выше 90 %, что подтверждает возможность ее применения для задач дефектоскопии.

Данное решение более пригодно в рамках автоматизации процесса контроля, но внедрение этой технологии ограничивается тем, что нейронная сеть представляет собой вероятностный метод решения задач. Поэтому эффективнее использовать ее для более прикладных задач, связанных с распознаванием дефектов. С учетом изложенного, предполагается увеличить объем используемых дефектограмм, а также продолжить исследования в части не только обнаружения дефектов, но и определения их типа.

Литература

1. Кретов, Е. Ф. Ультразвуковая дефектоскопия в энергомашиностроении. — 4-е изд., перераб. — Санкт-Петербург: СВЕН, 2014. — 312 с. — (Методы и средства неразрушающего контроля) (Библиотека «В мире неразрушающего контроля).

2. ГОСТ Р 55724-2013. Контроль неразрушающий. Соединения сварные. Методы ультразвуковые = Non-Destructive Testing. Welded Joints. Ultrasonic Methods: национальный стандарт Российской Федерации: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 8 ноября 2013 года № 1410-ст: дата введения 2015-07-01. — Москва: Стандартинформ, 2014. — 44 с.

3. Назаренко, С. Ю. Применение искусственных нейронных сетей в радиационном неразрушающем контроле / С. Ю. Назаренко, В. А. Удод // Дефектоскопия. 2019. № 6. С. 53–64. DOI: 10.1134/S013030821906006X.

4. Ревякин, А. М. Подходы к разработке системы распознавания для решения задачи определения контента цифровых изображений / А. М. Ревякин, А. В. Скурнович // Интернет-журнал «Науковедение». 2016. Т. 8, № 4. Статья № 30TVN416. 14 с. URL: http://naukovedenie.ru/PDF/30TVN416.pdf (дата обращения 06.06.2022).

5. Mask R-CNN: архитектура современной нейронной сети для сегментации объектов на изображениях // NanoNewsNet — Сайт о нанотехнологиях № 1 в России. — 2018. — 27 августа. URL: http://www.nanonewsnet.ru/news/2018/mask-r-cnn-arkhitektura-sovremennoi-neironnoi-seti-dlya-segmentatsii-obektov-na-izobrazhen (дата обращения 06.06.2022).

6. Bochkovskiy, A. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao // ArXiv. 2020. Vol. abs/2004.10934. 17 p. URL: http://arxiv.org/abs/2004.10934 (дата обращения 06.06.2022).

7. mAP (mean Average Precision) в детекции объектов // Neti ML — Machine Learning Neti. Умная видеоаналитика. URL: http://ml.i-neti.ru/map-mean-average-precision (дата обращения 06.06.2022).

8. Принцип и реализация немаксимального подавления NMS (Non-Maximum Suppression) // Русские Блоги. URL: http://russianblogs.com/article/79711224579 (дата обращения 06.06.2022).

9. Tan, M. EfficientDet: Scalable and Efficient Object Detection / M. Tan, R. Pang, Q. V. Le // Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020) (Seattle, WA, USA, 13–19 June 2020). — Institute of Electrical and Electronics Engineers, 2020. — Pp. 10778–10787. DOI: 10.1109/CVPR42600.2020.01079.

10. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection / S. Zhang, C. Chi, Y. Yao, [et al.] // Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020) (Seattle, WA, USA, 13–19 June 2020). — Institute of Electrical and Electronics Engineers, 2020. — Pp. 9756–9765. DOI: 10.1109/CVPR42600.2020.00978.

11. Liu, S. Learning Spatial Fusion for Single-Shot Object Detection / S. Liu, D. Huang, Y. Wang // ArXiv. 2019. Vol. abs/1911.09516. 10 p. DOI: 10.48550/arXiv.1911.09516.

12. Lee, Y. CenterMask: Real-Time Anchor-Free Instance Segmentation / Y. Lee, J. Park // Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020) (Seattle, WA, USA, 13–19 June 2020). — Institute of Electrical and Electronics Engineers, 2020. — Pp. 13903–13912. DOI: 10.1109/CVPR42600.2020.01392.

13. Расстояние Махаланобиса // Портал знаний StatSoft. URL: http://statistica.ru/theory/rasstoyanie-makhalonobisa/ (дата обращения 06.06.2022).

14. Kálmán, R. E. A New Approach to Linear Filtering and Prediction Problems // Journal of Basic Engineering. 1960. Vol. 82, Is. 1. Pp. 35–45. DOI: 10.1115/1.3662552.

DOI: 10.24412/2413-2527-2022-230-41-47

Application of YOLO Neural Network for Defect Recognition

E. A. Breht, PhD V. N. Konshina
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
ed.breht@mail.ru

Abstract. **The article considers an example of using the YOLO neural network to recognize and visualize defects in an image obtained in the process of ultrasonic testing by time-of-flight diffraction method. Of the 110 defect-grams containing images of defects, 100 of them were used to train the neural network, 10 - for verification. The possibility of recognition of defects by a neural network with sufficient accuracy for everyday practice of non-destructive testing is shown.**

Keywords: **neural network, ultrasonic testing, defectograms, recognition.**

REFERENCES

1. Kretov E. F. Ultrasonic flaw detection in power engineering [Ultrazvukovaya defektoskopiya v energomashinostroenii]. St. Petersburg, SVEN Publishing House, 2014, 312 p.

2. GOST R 55724-2013. Non-Destructive Testing. Welded Joints. Ultrasonic Methods [GOST R 55724-2013. Kontrol nerazrushayushchiy. Soedineniya svarnye. Metody ultrazvukovye]. Effective from July 01, 2015. Moscow, StandartInform Publishing House, 2014, 44 p.

3. Nazarenko S. Yu., Udod V. A. Application of Artificial Neural Networks in Radiation Non-Destructive Testing [Primenenie iskusstvennykh neyronnykh setey v radiatsionnom nerazrushayushchem kontrole], *Nondestructive Testing [Defektoskopiya]*, 2019, No. 6, Pp. 53–64.
DOI: 10.1134/S013030821906006X.

4. Revyakin A. M., Skurnovich A. V. Approaches to the Development of a Recognition System to Solve the Problem of Determining the Content of Digital Images [Podkhody k razrabotke sistemy raspoznavaniya dlya resheniya zadachi opredeleniya kontenta tsifrovykh izobrazheniy], *Online Journal Naukovedenie [Internet-zhurnal «Naukovedenie»]*, 2016, Vol. 8, No. 4. Art. No. 30TVN416. 14 p. Available at: http://naukovedenie.ru/PDF/30TVN416.pdf (accessed 06 June 2022).

5. Mask R-CNN: Modern Neural Network Architecture for Object Segmentation in Images [Mask R-CNN: arkhitektura sovremennoy neyronnoy seti dlya segmentatsii obektov na izobrazheniyakh], *NanoNewsNet — Nanotechnologies News from Russia [NanoNewsNet — Sayt o nanotekhnologiyakh № 1 v Rossii]*. Published online at August 27, 2018. Available at: http://www.nanonewsnet.ru/news/2018/mask-r-cnn-arkhitektura-sovremennoi-neironnoi-seti-dlya-segmentatsii-obektov-na-izobrazhen (accessed 06 June 2022).

6. Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection, *ArXiv*, 2020, Vol. abs/2004.10934, 17 p. Available at: http://arxiv.org/abs/2004.10934 (accessed 06 June 2022).

7. mAP (mean Average Precision) in Object Detection [mAP (mean Average Precision) v detektsii obektov], *Neti ML — Machine Learning Neti. Smart VideoAnalytics [Neti ML — Machine Learning Neti. Umnaya videoanalitika.]*. Available at: http://ml.i-neti.ru/map-mean-average-precision (accessed 06 June 2022).

8. The Principle and Implementation of Non-Maximum Suppression NMS [Printsip i realizatsiya nemaksimalnogo podavleniya NMS (Non-Maximum Suppression)], *Russian Blogs [Russkie Blogi]*. Available at: http://russianblogs.com/article/79711224579 (accessed 06 June 2022).

9. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and Efficient Object Detection, *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, June 13–19, 2020*. Institute of Electrical and Electronics Engineers, 2020, Pp. 10778–10787. DOI: 10.1109/CVPR42600.2020.01079.

10. Zhang S., Chi C., Yao Y., et al. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection, *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, June 13–19, 2020*. Institute of Electrical and Electronics Engineers, 2020, Pp. 9756–9765. DOI: 10.1109/CVPR42600.2020.00978.

11. Liu S., Huang D., Wang Y. Learning Spatial Fusion for Single-Shot Object Detection, *ArXiv*, 2019, Vol. abs/1911.09516, 10 p. DOI: 10.48550/arXiv.1911.09516.

12. Lee Y., Park J. CenterMask: Real-Time Anchor-Free Instance Segmentation, *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, June 13–19, 2020*. Institute of Electrical and Electronics Engineers, 2020, Pp. 13903–13912.
DOI: 10.1109/CVPR42600.2020.01392.

13. Mahalanobis Distance [Rasstoyanie Makhalanobisa], *StatSoft Knowledge Portal [Portal znaniy StatSoft]*. Available at: http://statistica.ru/theory/rasstoyanie-makhalonobisa (accessed 06 June 2022).

14. Kálmán R. E. A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering*, 1960, Vol. 82, Is. 1, Pp. 35–45. DOI: 10.1115/1.3662552 (accessed 06 June 2022).

DOI: 10.24412/2413-2527-2022-230-48-52

Реализация анализа главных компонентов с RHadoop

Сунь Ифань
Петербургский государственный университет путей сообщения Императора Александра I
Нанкин, Китай
syf105736@gmail.com

к.воен.н. Р. Г. Гильванов
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
gilvanov1950@mail.ru

Аннотация. **Hadoop — это очень популярная распределенная архитектура обработки больших данных, которая использует HDFS и MapReduce для высокоскоростных вычислений и хранения массивных данных. R — это широко используемое программное обеспечение с открытым исходным кодом с расширенными функциями статистического анализа данных. В этой статье RHadoop использовался для разработки функции анализа главных компонентов, которая может выполнять распределенные параллельные вычисления, и для выполнения анализа главных компонентов набора данных «iris» для проверки осуществимости. Использование RHadoop значительно повышает практичность и эффективность выполнения языка R для обработки больших данных, а также может сэкономить затраты на обработку больших данных языком R.**

Ключевые слова: **язык R, Hadoop, RHadoop, большие данные, анализ данных, MapReduce, HDFS.**

Введение

С наступлением эры облачных данных большие данные становятся все более и более заинтересованными и используемыми. После многих лет накопления пользовательских сетевых данных и эксплуатационных данных на некоторых предприятиях единица измерения этих данных достигла PB (петабайт) или даже ZB (зеттабайт) [1]. В этом контексте Hadoop широко используется в качестве распределенной файловой системы с открытым исходным кодом и модели параллельных вычислений [2]. Язык R имеет хорошее совершенство в статистическом анализе [3]. RHadoop является мостом между языком R и Hadoop. RHadoop чрезвычайно популярен в современных распределенных архитектурах больших данных, обеспечивая хранение больших данных и высокоскоростные вычисления с помощью MapReduce и HDFS.

Hadoop

Двумя ядрами Hadoop являются распределенная файловая система HDFS и распределенный параллельный фреймворк MapReduce, которые используются для хранения и обработки больших данных соответственно [4]. Hadoop — это распределенная вычислительная платформа с открытым исходным кодом, принадлежащая Apache. Она может поддерживать несколько языков программирования [5], обладает высокой надежностью, эффективностью (тысячи компьютеров могут быть вычислены вместе) и высокой масштабируемостью. Кроме того, она обладает низкой стоимостью (можно использовать обычный персональный компьютер). Основываясь на двух его ядрах, пользователи могут легко организовать компьютерные ресурсы для создания собственной распределенной вычислительной платформы; они также могут использовать хранилище и вычислительные мощности кластера для завершения обработки больших данных.

Распределенная файловая система HDFS. Ее полное название — Hadoop Distributed File System [5]. По своей сути HDFS является узлом имени (главный узел) и узлом данных (подчиненный узел), а также блоком данных [6]. В кластере узел имени необходим для управления всем другим хранилищем узла данных (администратор всей HDFS), а узел данных отвечает только за хранение данных. При хранении HDFS-файл разделяется на несколько блоков данных, которые также хранятся отдельно в нескольких узлах данных и записываются в узел имени.

Режим хранения HDFS показан на рисунке 1. Целью этого является поддержка крупномасштабного хранения данных и снижение потребления узлов при адресации.

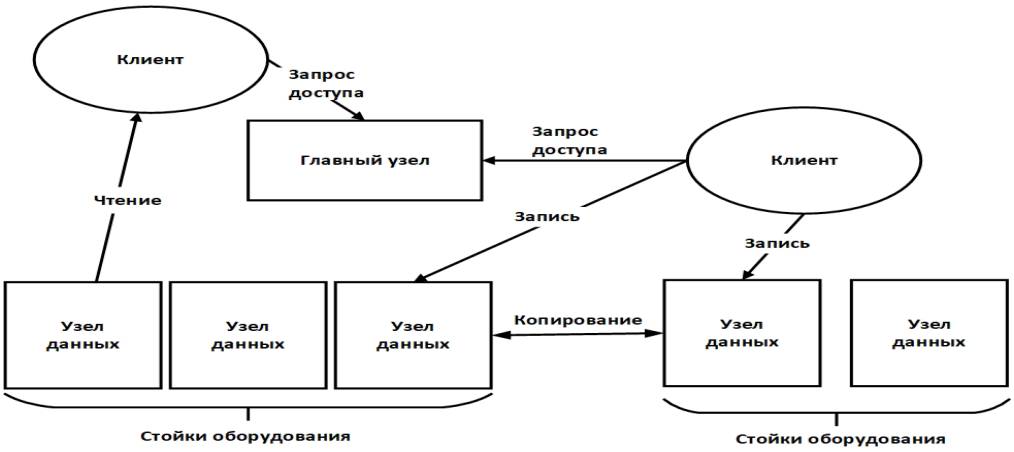


Рис. 1. Режим хранения распределенной файловой системы

Распределенный параллельный фреймворк MapReduce. MapReduce — это фреймворк распределенного параллельного программирования [7]. Именно через кластер через несколько машин одновременно происходит параллельная обработка больших данных.

MapReduce сначала загрузит файл из HDFS и предварительно обработает файл (Split), а затем преобразует данные в <key, value> пару «ключ — значение» [8], подходящую для задачи сопоставления через RR (RecordReader), и введет ее в задачу Map. Задача карты выводит <key, value> как промежуточные результаты в соответствии с пользовательскими правилами; эти результаты обрабатываются Shuffle и отправляются в задачу Reduce в качестве входных данных [5]. Задача Reduce выполняет пользовательскую логику, которая выводит результат в модуль OutputFormat. Модуль OutputFormat проверяет выходные данные, если условия выполнены, выводит их в HDFS для хранения. Более интуитивно понятный процесс показан на рисунке 2.

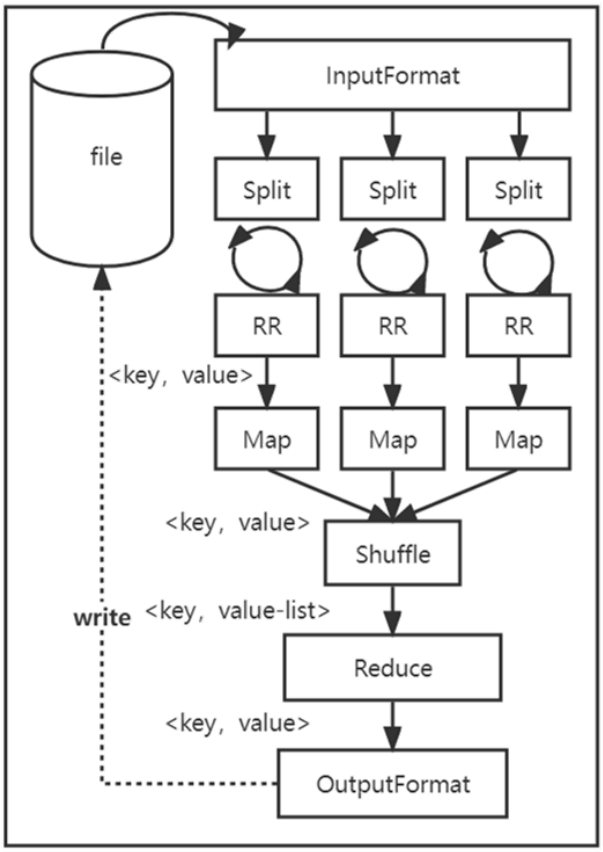


Рис. 2. Рабочий процесс MapReduce

R и RHadoop

R является платформой с открытым исходным кодом и свободным статистическим программным обеспечением, и многие методы статистического анализа могут быть легко завершены на R [9]. Недостатком языка R является то, что он не может эффективно обрабатывать большой объем данных, потому что язык R будет загружать данные в память при анализе данных, что также является сложной точкой в развитии языка R.

RHadoop представляет собой комбинацию языка R и Hadoop, разработанных Revolution Analytics, и кода с открытым исходным кодом для сообщества Github. RHadoop содержит три пакета R (rmr2, rhdfs, rhbase), которые соответствуют частям MapReduce, HDFS и HBase архитектуры системы Hadoop [10]. Главное дело, RHadoop просто предоставляет API (интерфейсы прикладного программирования), которые используют Hadoop. Пользователям RHadoop необходимо только иметь общее представление об архитектуре Hadoop для проектирования параллельных вычислений. Архитектура RHadoop показана на рисунке 3.

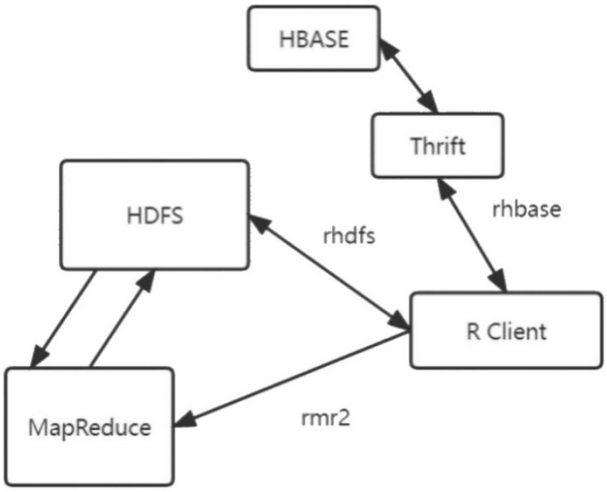


Рис. 3. Архитектура RHadoop

rmr2 предоставляет самый простой и эффективный способ написания задач MapReduce [11]. В плане объема программирования он намного меньше, чем Java, и в то же время имеет множество преимуществ, что обеспечивает удобство для последующего обслуживания. rhdfs предоставляет метод манипулирования данными HDFS, в то время как rhbase предоставляет метод манипулирования данными HBase.

С помощью этих пакетов язык R может использовать возможности Hadoop для эффективной обработки больших данных.

Структура сценария R с использованием RHadoop

Существует фиксированный способ использования RHadoop для задач MapReduce, который вызывает пользовательские функции map и reduce [12]. Способ работы RHadoop позволяет пользователям определять функции map и reduce. Сценарий для использования RHadoop показан ниже:

```
library(rmr2)
map<-function(k,v) { ...}
reduce<-function(k,vv) { ...}
mapreduce(
input ="data.txt",
output="output",
textinputformat =rawtextinputformat,
map = map,reduce=reduce)
```

Сначала нужно загрузить пакет rmr2, затем определить функцию map и использовать <key, value> в качестве входных данных для функции [12]. Следующим шагом требуется определить <key, value-list> в качестве входных данных для функции reduce. Наконец, настроить и запустить

задачу MapReduce. Таким образом задачи MapReduce могут быть запущены, а окончательные результаты будут сохранены в HDFS. Для просмотра результатов нужно только использовать функцию form.dfs(/…) в клиенте языка R.

Поскольку MapReduce может получить доступ только к файловой системе HDFS, перед использованием функции MapReduce необходимо сохранить данные в файловой системе HDFS.

Дизайн анализа главных компонентов

Анализ главных компонентов является одним из наиболее важных методов анализа данных в языке R, но использование языка R не может выполнять анализ главных компонентов больших данных. С помощью RHadoop анализ главных компонентов может выполняться в режиме MapReduce для достижения анализа больших данных.

Основная идея анализа главных компонентов заключается в использовании наиболее важных аспектов данных вместо исходных данных [13]. Он может сохранять основную информацию исходных данных и уменьшать размерность.

Процесс анализа главных компонентов выглядит следующим образом:

1. Стандартизировать образцы.
2. Рассчитать матрицу коэффициентов корреляции выборки.
3. Рассчитать собственные значения и собственные векторы матрицы коэффициентов корреляции.
4. Рассчитать ставку взноса на отклонение и кумулятивную ставку взноса на отклонение.
5. Выполнить визуальные иллюстрации.

Стандартизируем образцы. Структура кода:

```
data_mean=function(x){
 col_mean=function(x){
  map1=function(.,v){
   keyval(1:2,list(colSums(v),dim(v)[1]))
  }
  reduce1=function(k,v){
   keyval(k,list(Reduce("+",v)))
  }
  a.mean = values(from.dfs(mapreduce(input=x,map = map1,
       reduce = reduce1,combine = T)))
  b.mean = a.mean[[1]]/a.mean[[2]]
 }
 col_meandata=col_mean(x)
 map2=function(.,v){
  keyval(1,list(scale(v,center = col_meandata,scale = F)))
 }
 reduce2=function(k,v){
  keyval(1,list(Reduce("+",v)))
 }
 c.mean=values(from.dfs(mapreduce(input=x, map = map2,
       reduce=reduce2,combine = T)))
}
```

В этой части кода есть две основные части. Суммируем столбцы и используем результаты для следующего этапа стандартизации.

Рассчитаем матрицу коэффициентов корреляции выборки. Структура кода:

```
data_cm=function(x){
 data_st=data_mean(x)
 map3=function(.,v){
  keyval(1,list(cor(data_st[[1]])))
 }
 reduce3=function(k,v){
  keyval(1,list(Reduce("+",v)))
 }
 values(from.dfs(mapreduce(input=x,map=map3,
reduce = reduce3,combine = T)))
}
```

Эта часть кода в основном использует функцию cor() для вычисления матрицы коэффициента корреляции. Наконец, возвращается значение val.

Рассчитаем собственные значения и собственные векторы матрицы коэффициентов корреляции. Структура кода:

```
data_ei=function(x){
 cm=data_cm(x)[[1]]
map4=function(.,v){
 keyval(1:2,list(eigen(cm)$val,eigen(cm)$vec))
}
reduce4=function(k,v){
 keyval(k,list(Reduce("+",v)))
}
values(from.dfs(mapreduce(input = x,map=map4,
reduce = reduce4,combine = T)))
}
```

Дизайн кода для всего шаблона MapReduce завершен. Благодаря возвращаемому значению этой функции, мы можем использовать статистические функции на языке R для следующей части обработки анализа данных.

Мы также можем объединить код для всего процесса. Рекомендуется задействовать пользовательские функции для вычисления собственных значений и собственных векторов больших данных.

Удалим последний столбец данных «iris» для тестирования. Результаты теста согласуются с анализом, проведенным на языке R. Результаты теста показаны на рисунке 4.

```
> j
[[1]]
[1] 3.25012182 0.52244098 0.14063067 0.08680653

[[2]]
           [,1]       [,2]       [,3]        [,4]
[1,] -0.5012166  0.4611031  0.6935302 -0.23490782
[2,] -0.4941183  0.5122528 -0.7023221 -0.01370718
[3,] -0.4812482 -0.6424648 -0.1186062 -0.58444139
[4,] -0.5225219 -0.3349926  0.1081301  0.77656853
```

Рис. 4. Результаты теста

Рассчитаем ставку взноса на отклонение и кумулятивную ставку взноса на отклонение. Структура кода:

```
Pro <- j[[1]]/sum(j[[1]])
Cum <- cumsum(Pro)
```

Для наглядности приведем графическую иллюстрацию — результат вычислений показан на рисунке 5. Структура кода:

```
plot(j[[1]],type="b",
    cex=2,
    cex.lab=2,
    cex.axis=2,
    lty=2,
    lwd=2,
    xlab = "num",
    ylab="var")
```

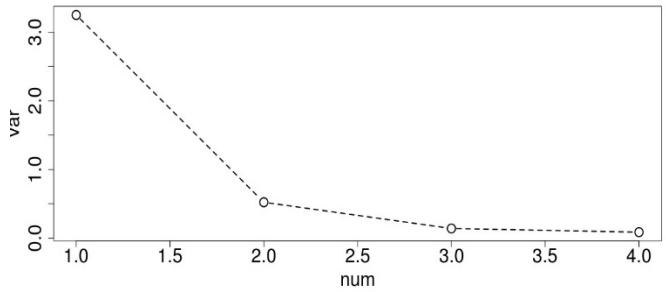


Рис. 5. Объясненная вариация

На рисунке 5 показано, что требуется только 2 главных компонента. Совокупная ставка взноса превысила 90 %. Согласно приведенным результатам анализа, такой метод анализа главных компонентов также может обеспечить эффективное параллельное вычисление на кластерах RHadoop.

Заключение

Изменения в информационных технологиях привели к появлению больших данных, которые сами по себе являются продуктом материи и капитала, поэтому можно сказать, что использование больших данных станет важным фактором повышения конкурентоспособности, производительности, инноваций и стимулирования прибыли. Поэтому анализ больших данных также стал самой передовой технологией. С наступлением эры больших данных использование архитектуры распределенной системы Hadoop для чтения, хранения и обработки больших данных стало одним из решений. Сочетание R и Hadoop значительно улучшило возможности статистического анализа больших наборов данных и позволило получить больше вычислительных ресурсов, так что данные могут использоваться более эффективно, полезная информация может быть получена из данных, а скорость обработки данных также принесла больше экономических выгод. В статье проверяется возможность реализации анализа главных компонентов для больших данных с помощью RHadoop.

Литература

1. Чехарин, Е. Е. Большие данные: большие проблемы // Перспективы науки и образования. 2016. № 3 (21). С. 7–11.

2. Дзидзава, Э. Т. Большие данные и Hadoop: обзорный доклад / Э. Т. Дзидзава, К. М. Ахмедов // Вестник магистратуры. 2021. № 1-1 (112). С. 30–32.

3. Venables, W. N. An Introduction to R. Second Edition / W. N. Venables, D. M. Smith; R Development Core Team. — Godalming: Network Theory Ltd., 2009. — 144 p.

4. White, T. Hadoop. The Definitive Guide: Storage and Analysis at Internet Scale. Fourth Edition. — Sebastopol (CA): O'Reilly Media Inc., 2015. — 754 p.

5. Lin, Z.-Y. Big data technology principle and application concept: Storage. Processing. Analysis and application. Third Edition (in Chinese). — Beijing: People's Posts and Telecommunications Publishing House, 2020. — 302 p. — (Big Data Innovation Talent Training Series).

6. Гусейнов, А. А. Исследование распределенной обработки данных на примере системы Hadoop / А. А. Гусейнов, И. А. Бочкова // Актуальные проблемы авиации и космонавтики: Материалы XII Международной научно-практической конференции, посвященной Дню космонавтики (Красноярск, Россия, 11–15 апреля 2016 г.): в 2 тт. / под общ. ред. Ю. Ю. Логинова. — Красноярск: Сибирский гос. аэрокосмический ун-т им. М. Ф. Решетнева, 2016. — Т. 1. — С. 606–608.

7. Гладкий, М. В. Модель распределенных вычислений MapReduce // Труды БГТУ. Физико-математические науки и информатика. 2016. № 6 (188). С. 194–198.

8. Dean, J. MapReduce: Simplified Data Processing on Large Clusters / J. Dean, S. Ghemawat // Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI '04) (San Francisco, CA, USA, 06–08 December 2004). — Berkeley (CA): USENIX Association, 2004. — Pp. 137–149.

9. Adler, J. R in a Nutshell: A Desktop Quick Reference. Second Edition. — Sebastopol (CA): O'Reilly Media Inc., 2012. — 724 p.

10. Prajapati, V. Big Data Analytics with R and Hadoop. — Birmingham: Packt Publishing, 2013. — 238 p.

11. Prathibha, U. Big Data Analysis with R Programming and RHadoop / U. Prathibha, M. Thillainayaki, A. Jenneth // International Journal of Trend in Scientific Research and Development. 2018. Vol. 2, Is. 4. Pp. 2623–2627. DOI: 10.31142/ijtsrd15705.

12. Oancea, B. Integrating R and Hadoop for Big Data Analysis / B. Oancea, R. M. Dragoescu // Revista Română de Statistică. 2014. Nr. 2. Pp. 83–94.

13. Kabacoff, R. I. R in Action, Third Edition: Data Anal-ysis and Graphics with R and Tidyverse. — Shelter Island (NY): Manning, 2022. — 656 p.

DOI: 10.24412/2413-2527-2022-230-48-52

Implementing Principal Component Analysis with RHadoop

Sun Yifan
Emperor Alexander I St. Petersburg
State Transport University
Nanjing, China
syf105736@gmail.com

PhD R. G. Gilvanov
Emperor Alexander I St. Petersburg
State Transport University
Saint Petersburg, Russia
gilvanov1950@mail.ru

Abstract. **Hadoop is a very popular distributed big data architecture that uses HDFS and MapReduce for high-speed computation and massive data storage. R is a widely used open source software with advanced statistical data analysis features. In this paper, RHadoop has been used to develop a principal component analysis function that can perform distributed parallel computations and to perform principal component analysis on an iris dataset to check feasibility. Using RHadoop significantly improves the usability and efficiency of R language execution for big data processing, and can also save the cost of big data processing in the R language.**

Keywords: **R language, Hadoop, RHadoop, big data, data analysis, MapReduce, HDFS.**

REFERENCES

1. Chekharin E. E. Big Data: Big Problems [Bolshie dannye: bolshie problemy], *Perspectives of Science and Education [Perspektivy nauki i obrazovaniya]*, 2016, No. 3 (21), Pp. 7–11.

2. Dzidzava E. T., Akhmedov K. M. Big Data and Hadoop: Overview Report [Bolshie dannye i Hadoop: obzornyy doklad], *Bulletin of the Magistracy [Vestnik magistratury]*, 2021, No. 1-1 (112), Pp. 30–32.

3. Venables W. N., Smith D. M. An Introduction to R. Second Edition. Godalming, Network Theory Ltd., 2009, 144 p.

4. White T. Hadoop. The Definitive Guide: Storage and Analysis at Internet Scale. Fourth Edition. Sebastopol (CA), O'Reilly Media Inc., 2015, 754 p.

5. Lin Z.-Y. Big data technology principle and application Concept: Storage. Processing. Analysis and application. Third Edition (in Chinese). Beijing, People's Posts and Telecommunications Publishing House, 2020, 302 p.

6. Guseinov A. A., Bochkova I. A. Research of the Distributed Data Processing on the Example of the Hadoop System [Issledovanie raspredelennoy obrabotki dannykh na primere sistemy Hadoop], *Topical Issues in Aeronautics and Astronautics: Proceedings of the XII International Scientific and Research Conference [Aktualnye problemy aviatsii i kosmonavtiki: Materialy XII Mezhdunarodnoy nauchno-prakticheskoy konferentsii, posvyashchennoy Dnyu kosmonavtiki], Krasnoyarsk, Russia, April 11–15, 2016. Volume 1.* Kranoyarsk, Reshetnev Siberian State Aerospace University, 2016, Pp. 606–608.

7. Gladkiy M. V. The Model of Distributed Computing MapReduce [Model raspredelennykh vychisleniy MapReduce], *Proceedings of BSTU. Physics and Mathematics. Informatics [Trudy BGTU. Fiziko-matematicheskie nauki i informatika]*, 2016, No. 6 (188), Pp. 194–198.

8. Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters, *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI '04), San Francisco, CA, USA, December 06–08, 2004*. Berkeley (CA), USENIX Association, 2004, Pp. 137–149.

9. Adler J. R in a Nutshell: A Desktop Quick Reference. Second Edition. Sebastopol (CA), O'Reilly Media Inc., 2012, 724 p.

10. Prajapati V. Big Data Analytics with R and Hadoop. Birmingham, Packt Publishing, 2013, 238 p.

11. Prathibha U., Thillainayaki M., Jenneth A. Big Data Analysis with R Programming and RHadoop, International Journal of Trend in Scientific Research and Development, 2018, Vol. 2, Is. 4, Pp. 2623–2627. DOI: 10.31142/ijtsrd15705.

12. Oancea B., Dragoescu R. M. Integrating R and Hadoop for Big Data Analysis, Romanian Statistical Review, 2014, No. 2, Pp. 83–94.

13. Kabacoff R. I. R in Action, Third Edition: Data Analysis and Graphics with R and Tidyverse. Shelter Island (NY), Manning, 2022, 656 p.

DOI: 10.24412/2413-2527-2022-230-53-58

Разработка кросс-платформенных приложений на языке Python и фреймворке Kivy

А. И. Федотова, к.воен.н. Р. Г. Гильванов
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
sasha.fedotova.01@mail.ru, gilvanov1950@mail.ru

Аннотация. **В статье рассматриваются варианты разработки приложений, такие как Web-приложения, нативные и кросс-платформенные приложения. Рассматриваются достоинства и недостатки каждого варианта. Сравнивается нативная и кросс-платформенная разработка. Подробнее рассматривается вариант кросс-платформенной разработки приложений, популярные фреймворки. В статье дается обзор языка программирования Python, приведены его достоинства и недостатки. Рассмотрена возможность кросс-платформенной разработки на языке программирования Python с помощью фреймворка Kivy.**

Ключевые слова: **разработка приложений, web-приложения, нативные приложения, кросс-платформенные приложения, фреймворки кросс-платформенной разработки, язык программирования Python, фреймворк Kivy.**

Введение

В современном мире существует большое многообразие техники, с которой взаимодействует человек, и соответственно которой нужен графический интерфейс. Эти устройства могут работать под множеством разнообразных операционных систем: Windows, Linux, MacOS, Android, iOS.

Существует множество различных приложений (как на мобильные устройства, так и на компьютеры, ноутбуки), которые решают большое количество разнообразных задач. Например, это приложения для заметок, планирования, разнообразные игры, приложения для обучения, прослушивания музыки, просмотра видео, приложения для доступа к различным социальным сетям, банкам и т.п. Эти приложения постоянно обновляются, а также разрабатываются новые приложения.

Пользователи этих приложений могут использовать устройства с разными операционными системами. Также, часто пользователю хочется иметь доступ к предоставляемым услугам в приложении с нескольких своих устройств. Например, на ноутбуке (с операционной системой Windows), на планшете (с Android) и на телефоне (с iOS), на всех устройствах желателен доступ к приложению.

Таким образом, перед разработчиком стоит вопрос выбора реализации приложения. Обычно рассматриваются такие варианты как, web-приложение, нативные приложения и кросс-платформенные приложения.

Web-приложение

Web-приложение – это прикладное программное обеспечение, которое работает на web-сервере, в отличие от «стационарных» программ, которые запускаются локально в операционной системе устройства. Для доступа к web-приложению пользователю необходимо активное подключение к Интернету.

Любое web-приложение представляет собой набор статических и динамических web-страниц. Статическая web-страница — это страница, которая всегда отображается перед пользователем в неизменном виде. Web-сервер отправляет страницу по запросу web-браузера без каких-либо изменений. В противоположность этому, сервер вносит изменения в динамическую web-страницу перед отправкой ее браузеру. По причине того, что страница меняется, она называется динамической [1].

Несомненным достоинством web-приложений является их доступность с любого устройства с доступом в Интернет.

Однако это же означает что без доступа в сеть пользователь не может воспользоваться приложением, что является недостатком.

Также отметим, что зачастую на мобильных устройствах гораздо удобнее пользоваться «стационарными» приложениями.

Нативная разработка

Под «нативной разработкой» (англ. *native* — родной, естественный) подразумевается использование оригинальных инструментов, предлагаемых каждой платформой — Swift и Cocoa для iOS и Java или Kotlin с набором инструментов для разработки программного обеспечения (Software Development Kit, SDK) от Android Open Source Project (AOSP) для Android [2].

У нативной разработки есть неоспоримые преимущества, как например [3]:

1. Скорость работы приложения. Код получаемый в результате нативной разработки является оптимальным для своей операционной системы.

2. Полная поддержка со стороны магазинов приложения (App Store и Google Play). Магазины приложений могут просто не пропустить приложение, если качество картинки слишком низкое, скорость выполнения слишком медленная и т.п. Всего этого (качества изображения, скорости выполнения) легче достичь с помощью нативной разработки.

3. Использование последних технологий. Так как средства нативной разработки поддерживаются компаниями-производителями устройств и операционных систем, то весь функционал (аппаратный и программный), даже которые только был добавлен, становится доступен для нативных приложений.

Однако имеется и недостаток — сложность (или высокая стоимость). Для разработки уникального, качественного приложения требуется немало времени и высокая квалификация разработчика.

Кросс-платформенная разработка

Кросс-платформенная разработка — это способ создания приложения с возможностью адаптации под несколько операционных систем. Чаще всего кросс-платформенная разработка рассматривается для мобильных приложений (операционные системы Android и iOS). Однако существует также кросс-платформенная разработка и для десктопных (англ. Desktop – рабочий стол) приложений (операционные системы Windows, Linux и MacOS).

Достоинства кросс-платформенной разработки [3]:

1. Требуется меньше ресурсов для реализации приложения сразу под несколько платформ. Программистов, занимающихся проектом, нужно ровно в два раза меньше. Дизайнер делает только один набор графики. Все это снижает количество рабочих часов и бюджет проекта.

2. Меньшее время на разработку. За счет отсутствия уникальных элементов интерфейса и более простых технологий разработки кросс-платформенных приложений, время на создание простых продуктов, как правило, меньше.

3. Упрощенный цикл обновления продукта. Если в проект нужно что-то добавить или исправить какую-то ошибку, это делается сразу для всех платформ, на которые распространяется проект.

4. Возможность использования мобильной версии сайта. В большинстве случаев языки для кросс-платформенной технологии разработки мобильных приложений входят в семейство языков JavaScript. Поэтому если у проекта уже есть мобильная версия сайта, значительная часть кода и материалов может быть использована в приложении без изменений.

5. Использование единой логики приложения. Логика, заложенная в работу приложения, будет работать гарантированно одинаково для всех платформ. Довольно часто это может являться и минусом из-за разной архитектуры операционных систем. Яркий пример — кнопка «Назад» в навигации между экранами. В Android предусмотрена аппаратная кнопка Back для этих целей, в iOS — движение пальцем от левой части экрана или же наличие кнопки в левой части навигационной панели. Если кнопку не делать вовсе, пользователи iOS не смогут вернуться назад. Если сделать, но не на том месте и выглядящую нестандартно, пользователям iOS будет непривычно и неудобно; а если сделать как в iOS, будет непривычно пользователям Android. Однако написанная и отлаженная один раз логика содержит потенциально меньшее количество ошибок и расхождений в своей работе. Поэтому вам не придется проделывать двойную и тройную работу по поиску проблем на каждой платформе.

Недостатки кросс-платформенных приложений:

1. Устаревание. С каждым обновлением операционной системы появляется возможность (необходимость) улучшить приложение, добавить в него новый, доступный функционал. Для нативной разработки это практически не составляет труда, потому средства нативной разработки почти сразу же предоставляют такую возможность. Однако при кросс-платформенной разработке необходимо дождаться обновления фреймворка, на котором написано кросс-платформенное приложение. А это может занять куда больший срок, по сравнению с нативными приложениями.

2. Низкая скорость. Зачастую кросс-платформенные приложения представляют собой веб-страницы, которые не особо расторопны, например, в скроллинге тяжелого контента: картинок, анимаций и т. д.

3. Трудности дизайна**.** Если вы хотите, чтобы вид вашего приложения соответствовал профессиональному и хорошо-проработанному системному дизайну каждой из платформ, будь то iOS или Android, вам придется разрабатывать дизайн для обеих операционных систем по отдельности. iOS и Android приложения имеют свои собственные, уникальные стандарты дизайна, а так как кросс-платформенное приложение не отвечает им, его вид придется «подгонять» под соответствующие рамки. Получается, по окончании работы вы получите только одно приложение, а времени и денег вы потратили как на два.

4. Незащищенность исходного кода**.** Одним из серьезных минусов кросс-платформенных приложений является их небезопасность. В то время как нативное приложение может быть зашифровано перед выходом в официальный магазин, кросс-платформенное приложение остается «голым». Поскольку в основе многих кросс-платформенных приложений лежит HTML-страница, то ничего не стоит посмотреть ее исходный код и понять, как работает само приложение.

Подводя итоги, можно сказать, что обычно кросс-платформенные фреймворки абстрагируют разработчиков от особенностей написания кода под отдельную платформу. Однако, как только вопрос касается производительности или реализации специфического функционала под платформу, так сэкономленное при разработке время съедается на решении этих проблем.

Также стоит отметить, что подходы при разработке приложения можно комбинировать. Например, отрабатывать критичные к производительности экраны (лента новостей в социальной сети) на нативных технологиях, а второстепенные (экран профиля, экран настроек) — на кросс-платформенных.

Сравнение нативных и кросс-платформенных приложений

Как правило, выход любого бизнеса в Интернет протекает по следующему сценарию: сначала компания запускает сайт, затем его адаптируют под мобильные устройства, и если наблюдается прирост трафика, появляется смысл закрепиться среди владельцев мобильных гаджетов, и компания выпускает приложение.

Сравнивать мобильный сайт и приложение нет смысла: второе однозначно выигрывает за счет широты своих возможностей и отзывчивого интерфейса, взаимодействовать с которым через телефон или планшет гораздо комфортнее. Кроме того, приложение может работать без постоянного подключения к Интернету.

В таблице 1 приведено сравнение нативной и кросс-платформенной разработки по основным пунктам [5]. Из нее следует, что по возможности работать с конкретной операционной системой, качеству интерфейса и производительности нативная разработка абсолютно точно лучше кросс-платформенной. Однако для стартапов, маленьких проектов важнее скорость и стоимость разработки, а также возможность выпустить приложение сразу на несколько платформ, для привлечения большей аудитории. Рассмотрим кросс-платформенную разработку подробнее.

Таблица 1

Сравнение нативных и кросс-платформенных приложений

Критерий оценивания	Нативные приложения	Кросс-платформенные приложения
Стоимость разработки	Высокая стоимость разработки	Относительно низкая стоимость разработки
Возможности использования кода	Работает для одной платформы	Работает на нескольких платформах
Доступность интерфейсов устройств	SDK платформы обеспечивает беспрепятственный доступ ко всем интерфейсам устройства	Нет гарантированного доступа ко всем интерфейсам устройства
Согласованность пользовательского интерфейса	Согласованность с компонентами пользовательского интерфейса устройства	Ограниченная согласованность с компонентами пользовательского интерфейса устройства
Производительность	Безупречная производительность, поскольку приложение разработано для ОС устройства.	Высокая производительность, но нередки задержки и проблемы с совместимостью оборудования.

Популярные фреймворки для кросс-платформенной разработки

Существует множество решений для кросс-платформенной разработки. Наиболее популярные из них [6]:

- PhoneGap (JavaScript, HTML5 и CSS3);
- Ionic (HTML, CSS, JavaScript и Angular);
- React Native (JavaScript, Java, C++, Objective-C и Python);
- Flutter (Dart);
- Xamarin (C#).

Выбор конкретного фреймворка зависит во многом и от языка программирования, на котором хочет работать разработчик.

Одним из наиболее популярных языков программирования на данный момент является язык программирования Python. По рейтингу GitHub [7] последние 5 лет язык программирования Python не опускался ниже 2 места. По рейтингу TIOBE же Python занимает 1 место по состоянию на март 2022 [8]. Python — стабильный и распространенный язык. Он используется во многих проектах и в различных качествах: как основной язык программирования или для создания расширений и интеграции приложений. На Python реализовано большое количество проектов, также он активно используется для создания прототипов будущих программ. В данной статье будет рассмотрена возможность кросс-платформенной разработке на языке программирования Python.

Язык программирования Python

Python это высокоуровневый язык программирования общего назначения с динамической строкой типизацией и автоматический управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нем программ.

Примеры использования Python крупными компаниями [9]:

- **Google.** С момента появления языка компания взяла на вооружение лозунг «Python везде, где можем, а C ++ — где должны». Python не только является компонентом поискового движка, но и считается (наряду с C ++, Java и Go) одним из официальных серверных языков Google, приложения на которых разрешено развертывать в производственной среде.
- **Facebook**. Python занимает третье место (после C++ и Hack) среди самых популярных языков разработки, которыми пользуются инженеры технологического гиганта. На нем сделано более 5 000 коммитов для утилит и инфраструктурных приложений Facebook.
- **Instagram**. Платформа социальных сетей целиком создана на базе Python-фреймворка Django. Она ежедневно дает возможность 4 миллионам активных пользователей фотографировать, редактировать, делиться и сохранять свои творения в личном цифровом альбоме.
- **Spotify**. Крупный игрок на рынке и приложение для потоковой передачи музыки использует Python для аналитики данных. На ее основе работают алгоритмы рекомендаций в популярнейших функциях «Радио» и «Открытия недели».
- **Netflix**. Стриминговый сервис высоко оценил возможности стандартной библиотеки Python, чрезвычайно активное сообщество разработчиков и богатый выбор сторонних библиотек, доступных для решения практически любой конкретной проблемы. В своем блоге компания отмечала, что использует Python на протяжении всего жизненного цикла контента — от принятия решения о финансировании проектов, до управления сетью CDN, предоставляющей видео конечным пользователям.
- **Dropbox**. Популярное онлайн-хранилище применяет Python для оптимизации кода как серверной части, так и внешнего интерфейса. Для этой задачи они привлекали самого создателя «змеиного языка» Гвидо ван Розума. А в 2016 году Dropbox выпустили Pyston — свою собственную реализацию Python, совместимую с CPython и библиотекой NumPy.

Достоинства Python:

1. Удобство и простота.

Python легко читается и понимается. Синтаксис ядра языка минималистичен, за счет чего на практике редко возникает необходимость обращаться к документации.

2. Открытый исходный код.

Лицензия Python с открытым исходным кодом делает его легкодоступным, облегчает распространение и создание модификаций. Разработчики со всего мира могут бесплатно использовать язык и вносить свой вклад в его улучшение. К тому же, в случае с Python сами пользователи, а не крупные компании решают, как будет развиваться технология.

3. Встраиваемость и платформонезависимость.

Благодаря своей интерактивности и переносимости Python обладает хорошими возможностями для динамической семантики и быстрого прототипирования. Его можно легко встроить в широкий спектр приложений, даже в те, которые используют разные языки программирования. Поэтому с Python можно легко исправлять новые модули и расширять базовый словарный запас языка.

Python, как C++, Java и другие высокоуровневые языки программирования, может работать с разными типами компьютеров, ОС и баз данных практически без модификаций. Он хорошо интегрирован не только с популярными платформами Windows, Mac и Linux/UNIX, но и со встроенными системами, такими как Raspberry Pi и Gumstix. Программы на Python также позволяют реализовывать переносимые графические интерфейсы.

4. Большое количество фреймворков.

Одним из главных преимуществ языка Python является наличие у него большого числа фреймворков, упрощающих процесс разработки. Большинство фреймворков Python имеют четкую специализацию, в зависимости от типа и масштаба выполняемых с их помощью задач.

5. Динамическая типизация.

Python не знает тип переменной, пока код не запустится. Он автоматически назначает тип данных во время выполнения. Программисту не нужно заранее беспокоиться об объявлении переменных и их типов данных.

Недостатки Python:

1. Слабая поддержка многопроцессорности.

Многопроцессорность — важная часть написания приложения. Python поддерживает многопроцессорность, но из-за отсутствия прямой поддержки многопоточности (задачи выполняются параллельно в один поток), он может быть не таким гибким или удобным, как другие языки.

Это может создать определенные трудности при параллельном выполнении кода. Хотя подобные ограничения во многом снимается за счет многочисленных дополнительных библиотек Python, умеющих полноценно работать с многопоточностью.

2. Ограничение скорости.

Python часто критикуют за его скорость. Это интерпретируемый скриптовый язык, поэтому он работает относительно медленнее своих скомпилированных аналогов (например, C/C++ или Java), которым не нужно тратить время на перевод текста программы. Тем не менее, некоторые тесты на Python работают быстрее, чем на C и C++.

При этом, Python — не единственный, у кого есть потенциальные проблемы со скоростью. Ruby, Perl и даже JavaScript также находятся на более медленном конце «скоростной» шкалы.

3. Большая нагрузка на память.

Python — это язык, известный гибкостью подходов к типизации данных. Эта же динамическая типизация приводит к повышенному потреблению памяти. Поэтому Python будет неидеальным выбором для задач, интенсивно использующих память.

Исторически Python не был лучшим инструментом для написания GUI (Graphic User Interface) приложений. При разработке мобильных приложений стандартными языками считались Java (Android) и Swift (iOS).

Однако мобильная разработка на Python постепенно прогрессирует. Результатом этого прогресса являются несколько современных инструментов, один из них — фреймворк Kivy [10].

Фреймворк Kivy

Фреймворк Kivy был создан в 2011 году, это кросс-платформенный фреймворк, работающий на Windows, Mac, Linux и Raspberry Pi. В дополнение к стандартному вводу через клавиатуру и мышь он поддерживает мультитач, таким образом его можно использовать и для мобильной разработки Android и iOS. Kivy также поддерживает ускорение GPU своей графики, что во многом является следствием использования OpenGL ES2. У проекта есть лицензия MIT, поэтому библиотеку можно использовать бесплатно и вкупе с коммерческим программным обеспечением.

В настоящее время возможности фреймворка Kivy значительно расширена за счет библиотеки KivyMD. Аббревиатура MD означает Material Design (стандарт, созданный Google, которого нужно придерживаться при разработке приложений для Android и iOS) [11].

Во время разработки приложения через Kivy создается интуитивно понятный интерфейс — Natural User Interface, или NUI**.** Его главная идея в том, чтобы пользователь мог легко и быстро приспособиться к программному обеспечению без чтения инструкций.

Kivy не задействует нативные элементы управления, или виджеты. Все его виджеты настраиваются. Виджеты — это отображаемые на экране элементы управления, которыми пользователь может оперировать, например, кнопки, выпадающие списки и т. п. Это значит, что приложения Kivy будут выглядеть одинаково на всех платформах. Тем не менее, это также предполагает, что внешний вид приложения будет отличаться от нативных приложений пользователя. Это может стать как преимуществом, так и недостатком.

Для создания приложения Kivy необходимо создать подкласс App и переопределить метод build(). Данный метод можно считать главным для приложения, в него помещаются код интерфейса, или же в нем вызываются другие методы, определяющие интерфейс.

```
from kivy.app import App
from kivy.uix.label import Label

class MainApp(App):
    def build(self):
        // your App code

if __name__ == '__main__':
    app = MainApp()
    app.run()
```

Во фреймворке Kivy предусмотрено большое количество разнообразных виджетов, комбинируя которые можно создать нужный интерфейс приложения.

Для работы с дизайном приложения Kivy предоставляет свой собственный язык дизайна (разметки). В таком случае дизайн приложения должен быть описан в файле с расширением *.kv*. Это позволяет отделить логику приложения от его дизайна, то есть поддерживает принцип разделения ответственности, или модель Model-View-Controller (Модель-Представление-Контроллер).

Язык разметки Kivy очень прост и похож на язык разметки HTML.

```
<Button>:
    text: 'Press me'
    size_hint: (.5, .5)
    pos_hint: {'center_x': .5, 'center_y': .5}
    on_press: app.on_press_button()
```

Первая строка <Button> указывает на то, какой метод должен вызываться в коде Python для данного виджета. Далее устанавливается текст кнопки, его размер (ширина и высота), позиция и обработчик событий. Обработчик событий указывает на метод, который должен находиться в Python-классе приложения.

Таким образом, фреймворк Kivy представляет возможность разработки кросс-платформенного приложения с разделением логики и представления приложения.

По завершению разработки приложения можно собрать python-приложение для необходимых операционных систем из числа из поддерживаемых (Windows, Linux, MacOS, Raspberry Pi, Android и iOS). Для этого можно использовать различные инструменты, например, pyinstaller, buildozer и др.

Заключение

Выбор конкретных технологий для реализации приложения зависит от многих факторов, например, от того, для каких операционных систем и устройств необходимо разработать приложение, от необходимого функционала приложения, от субъективных предпочтений разработчика и от многого другого.

Нативная разработка предпочтительнее во многих ситуациях, но если Вы можете работать с не нативным набором инструментов пользовательского интерфейса, Вам нужно выйти сразу на несколько платформ, есть ограничение во времени и средствах, или же просто не хочется писать приложение на Java или Swift, то кросс-платформенная разработка станет отличным решением.

Существует множество фреймворков для кросс-платформенной разработки, и при выборе конкретного фреймворка также важную роль играют личные предпочтения, в том числе и в языке программирования.

Язык программирования Python и фреймворк Kivy — это отличный выбор для новичков в программировании, или для тех, кто просто предпочитает данный язык. А также для стартапов, для проектов, которые нужно вывести на рынок быстро и собрать аудиторию для определения дальнейшего развития продукта.

Литература

1. Общие сведения о веб-приложениях // Adobe. — Обновлено 21.05.2021. URL: http://helpx.adobe.com/ru/dreamweaver/using/web-applications.html (дата обращения 25.03.2022).

2. Льюис, Ш. Нативная разработка мобильных приложений: Перекрестный справочник для iOS и Android = Native Mobile Development: A Cross-Reference for iOS and Android / Ш. Льюис, М. Данн; пер. с англ. А. Н. Киселева. — Москва: ДМК-Пресс, 2020. — 376 с.

3. Мурзанаев, А. Технология создания мобильных приложений: нативная или кроссплатформенная разработка // Блог AppCraft. — 2021. — 18 марта. URL: http://appcraft.pro/blog/nativnaja_razrabotka (дата обращения 25.03.2022).

4. Нативные vs гибридные приложения // Блог UmbrellaIT. — 2017. — 03 августа. URL: http://umbrellait.com/ru/blog/native-vs-hybrid-app (дата обращения 25.03.2022).

5. Manchanda, A. The Ultimate Guide to Cross Platform App Development Frameworks in 2022 // Net Solutions. — Обновлено 04.09.2021. URL: http://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019 (дата обращения 25.03.2022).

6. Patel, A. Top 5 Cross-Platform App Development Frameworks You Need to Know in 2021 // Medium. — 2021. — 17 August. URL: http://medium.com/nerd-for-tech/top-5-cross-platform-app-development-frameworks-you-need-to-know-in-2021-405404cf12fd (дата обращения 20.03.2022).

7. GitHut 2.0. GitHub Languages Stats // GitHub. URL: http://madnight.github.io/githut/#/issues/2020/1 (дата обращения 20.03.2022).

8. TIOBE Index for March 2022 // TIOBE. URL: http://tiobe.com/tiobe-index (дата обращения 20.03.2022).

9. Разработка на Python — плюсы и минусы // Блог Eternalhost. — 2022. — 11 января. URL: http://eternalhost.net/blog/razrabotka/nedostatki-i-preimushhestva-python (дата обращения 13.02.2022).

10. Bader, D. Мобильная разработка на Python: обзор двух фреймворков // Proglib. Библиотека программиста. — 2018. — 16 февраля. URL: http://proglib.io/p/mobile-python (дата обращения 13.02.2022).

11. Постолит, А. Разработка кроссплатформенных мобильных и настольных приложений на Python: Практическое пособие. — Ridero, 2022. — 676 с.

DOI: 10.24412/2413-2527-2022-230-53-58

Development of Cross-Platform Applications with Python and Kivy Framework

A. I. Fedotova, PhD R. G. Gilvanov
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
sasha.fedotova.01@mail.ru, gilvanov1950@mail.ru

Abstract. **This article discusses application development options such as Web applications, native applications, and cross-platform applications. The advantages and disadvantages of each option are considered. Native and cross-platform development is compared. The option of cross-platform application development, popular frameworks is considered in more detail. The article provides an overview of the Python programming language, its advantages and disadvantages. The possibility of cross-platform development in the Python programming language using the Kivy framework is considered.**

Keywords: **application development, web applications, native applications, cross-platform applications, cross-platform development frameworks, Python programming language, Kivy framework.**

REFERENCES

1. General Information About Web Applications [Obshchie svedeniya o veb-prilozheniyakh], *Adobe*. Last update May 21, 2021. Available at: http://helpx.adobe.com/en/dreamweaver/using/web-applications.html (accessed 25 Mar 2022).

2. Lewis S., Dunn M. Native Mobile Development: A Cross-Reference for iOS and Android [Nativnaya razrabotka mobilnykh prilozheniy: Perekrestnyy spravochnik dlya iOS i Android]. Moscow, DMK Press, 2020, 376 p.

3. Murzanaev A. Technologies for Creating Mobile Applications: Native or Cross-Platform Development [Tekhnologiya sozdaniya mobilnykh prilozheniy: nativnaya ili krossplatformennaya razrabotka], *AppCraft*. Published online at March 18, 2021. Available at: http://appcraft.pro/blog/nativnaja_razrabotka (accessed 25 Mar 2022).

4. Native vs Hybrid Apps [Nativnye vs gibridnye prilozheniya]. *UmbrellaIT*. Published online at August 03, 2017. Available at: http://umbrellait.com/ru/blog/native-vs-hybrid-app (accessed 25 Mar 2022).

5. Manchanda A. The Ultimate Guide to Cross Platform App Development Frameworks in 2022, *Net Solutions*. Last update September 04, 2021. URL: http://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019 (accessed 25 Mar 2022).

6. Patel A. Top 5 Cross-Platform App Development Frameworks You Need to Know in 2021, *Medium*. Published online at August 17, 2021. Available at: http://medium.com/nerd-for-tech/top-5-cross-platform-app-development-frameworks-you-need-to-know-in-2021-405404cf12fd (accessed 20 Mar 2022).

7. GitHut 2.0. GitHub Languages Stats, *GitHub*. Available at: http://madnight.github.io/githut/#/issues/2020/1 (accessed 20 Mar 2022).

8. TIOBE Index for March 2022, *TIOBE*. Available at: http://tiobe.com/tiobe-index (accessed 20 Mar 2022).

9. Python Development — Pros and Cons [Razrabotka na Python — plyusy i minusy], *Eternalhost*. Published online at January 11, 2022. Available at: http://eternalhost.net/blog/razrabotka/nedostatki-i-preimushhestva-python (accessed 13 Feb 2022).

10. Bader D. Mobile Development with Python: An Overview of Two Frameworks [Mobilnaya razrabotka na Python: obzor dvukh freymvorkov], *Proglib. Programmer's Library [Proglib. Biblioteka programmista]*. Published online at February 16, 2018. Available at: http://proglib.io/p/mobile-python (accessed 13 Feb 2022).

11. Postolit A. Developing Cross-Platform Mobile and Desktop Applications with Python: A Practical Guide [Razrabotka krossplatformennykh mobilnykh i nastolnykh prilozheniy na Python: Prakticheskoe posobie]. Ridero, 2022, 676 p.