



ITS₂₀₂₀



Информационные технологии и системы

ИТС 2020

**Материалы
международной научной конференции**

**Materials
of the International Conference**

ITS 2020

Information Technologies and Systems

18 ноября 2020
БГУИР, Минск, Беларусь

18th November 2020
BSUIR, MINSK, BELARUS

iTechArt

software
technologies

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»



**Информационные технологии и системы 2020
(ИТС 2020)**

Материалы международной научной конференции
(БГУИР, Минск, Беларусь, 18 ноября 2020)

Information Technologies and Systems 2020
(ITS 2020)

Proceeding of The International Conference
(BSUIR, Minsk, Belarus, 18th November 2020)

Минск БГУИР 2020

УДК 004
ББК 32.81
И74

Редакционная коллегия:

Л. Ю. Шилин (главный редактор), А.А. Иванюк, С.В. Колосов, В.С. Муха, В.В. Голенков, А.В. Марков, Д.П. Кукин, А.А. Навроцкий, А.Ф. Трофимович, Я.Г. Гриневич, Д.В. Шункевич, А.Б. Гуринович (ответственный секретарь)

Информационные технологии и системы 2020 (ИТС 2020) =
И74 Information Tehnologies and Systems 2020 (ITS 2020) : материалы международной научной конференции, Минск, Беларусь, 18 ноября 2020 г. / Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2020
ISBN 978-985-543-013-2.

Сборник включает прошедшие рецензирование доклады международной научной конференции «Информационные технологии и системы 2020» (ИТС 2020).

Сборник предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере IT-технологий.

Материалы сборника одобрены организационным комитетом и печатаются в авторской редакции.

УДК 004
ББК 32.973.202

ISBN 978-985-543-013-2

Белорусский государственный университет
информатики и радиоэлектроники», 2020

ПРЕДИСЛОВИЕ

18 ноября 2020 года в городе Минске прошла международная научная конференция «Информационные технологии и системы» (ИТС 2020), организованная факультетом информационных технологий и управления учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», базовой организации государств-участников СНГ по высшему образованию в области информатики и радиоэлектроники.

В течение дня на пленарном и пяти секционных заседаниях было представлено 100 докладов по следующим направлениям:

Секция «Системы управления»:

- моделирование систем управления;
- автоматизированные исследования систем управления;
- проектирование систем управления;
- микропроцессорные системы управления;
- информационные технологии в системах управления;
- системы передачи и защиты дискретной информации;
- системы телемеханики;
- элементы и устройства систем управления.

Секция «Интеллектуальные информационные технологии»:

- проектирование баз знаний;
- интеллектуальный поиск информации;
- интеллектуальные решатели задач;
- пользовательские интерфейсы интеллектуальных систем.

Секция «Проектирование встраиваемых систем»:

- проектирование и быстрое прототипирование встраиваемых систем на ПЛИС;
- методологии проектирования IP-компонент встраиваемых систем;
- программная и аппаратная защита IP-компонент;
- контролепригодное проектирование и тестирование встраиваемых систем;
- реконфигурируемые встраиваемые системы.

Секция «Системы обработки информации»:

- математические модели и методы обработки информации;
- статистические, численные, логические, графовые, нейросетевые и другие;
- анализ многомерных данных;
- методологические проблемы и технологии подготовки специалистов в области систем обработки информации;
- технологии и стандарты программирования и проектирования систем обработки информации;
- системы и средства совместной работы пользователей;
- защита информации в автоматизированных системах обработки информации.

Секция «Математическое моделирование и компьютерная графика»:

- методы решения уравнений математической физики;
- методы нелинейной оптимизации;
- модели и методы радиофизики и приборов СВЧ;
- модели и методы тепло и массопереноса;
- математические модели и методы компьютерной графики;
- аппаратное обеспечение компьютерной графики;
- алгоритмы и методы интерактивных программных продуктов.

В работе конференции приняли участие учёные и специалисты Республики Беларусь, Российской Федерации, Казахстана.

Докладчики представляли как высшие учебные заведения, в их числе: БГУ, БГУИР, БНТУ, БГТУ, БГЭУ, БрГТУ, ПГУ, МИУ, Военная академия РБ, Гомельский государственный технический университета им. П. О. Сухого, Гродненский государственный университет им. Янки Купалы, ГГУ им. Ф. Скорины, Волжский государственный университет водного транспорта, ГУВПО «Белорусско-Российский Университет», Могилевский государственный университет продовольствия, Евразийский национальный университет имени Гумилёва – так и институты и коммерческие организации: НАН РБ, Объединенный институт проблем информатики НАН РБ, ИДСТУ СО РАН, ОАО «АГАТ-системы управления», ЕРАМ Systems.

Оргкомитет благодарит всех, кто счел возможным поделиться результатами своей деятельности и принял участие в работе конференции.

ORGANISING COMMITTEE

Prof.Dr.	Vadim Bogush	Rector of The Belarusian State University of Informatics and Radioelectronics
PhD.	Maxim Davydov	First Vice-Rector of BSUIR
Ph.D.	Anatoly Osipov	Vice-Rector for Research of BSUIR
Ph.D.	Victor Rybak	Vice-Rector for Education of BSUIR
Prof.Dr.	Leonid Shilin	Dean of The Faculty of Information Technologies and Control of BSUIR
	Dmitry Kuznetsov	Vice-rector for Personal Development of BSUIR

PROGRAMME COMMITTEE

Prof.Dr.	Vladimir Golenkov	Professor of Intellectual Information Technologies Department of BSUIR
Prof.Dr.	Vladimir Golovko	Head of Intellectual Information Technologies Department of BrSTU
Prof.Dr.	Alexander Ivaniuk	Professor of Computer Science Department of BSUIR
Ph.D.	Dmitry Kukin	Head of Computational Methods and Programming Department of BSUIR
Ph.D.	Alexander Markov	Head of Control Systems Department of BSUIR
Ph.D.	Anatoly Navrotsky	Head of Information Technologies in Automated Systems Department of BSUIR
Ph.D.	Vadim Zhuravlev	Head of Theoretical Bases of Electrical Engineering Department of BSUIR
Prof.Dr.	Stanislav Kolosov	Professor of Computational Methods and Programming Department of BSUIR
Ph.D.	Daniil Shunkevich	Head of Intellectual Information Technologies Department of BSUIR

Пленарные доклады

Plenary session

БИОИНСПИРИРОВАННЫЙ И ПСИХОАКУСТИЧЕСКИ МОТИВИРОВАННЫЙ АНАЛИЗ ЗВУКОВЫХ СИГНАЛОВ

Вашкевич М. И., Азаров И. С.

Кафедра электронных вычислительных средств,

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: vashkevich@bsuir.by, azarov@bsuir.by

Биоинспирированный и психоакустически мотивированные методы анализа звуковых сигналов являются основой многих систем, выполняющих обработку речевой и аудио информации. В основе этих методов лежит знание особенностей работы слуховой системы человека и особенностей восприятия им звуков. В данной работе производится обзор основных методов биоинспирированного и психоакустически мотивированного анализа звуковых сигналов. Дается краткое пояснения какие именно свойства и особенности слуховой системы описывает каждый метод. Среди биоинспирированных методов анализа описываются: гамматон и гаммачирп банки фильтров, roex-фильтры, а также модель слуховых фильтров Кейтса. Среди психоакустически мотивированных способов анализа приводятся метод ДПФ с суммированием энергий в субполосах, банки фильтров на основе вейвлет преобразования, а также банки фильтров на основе фазового преобразования.

ВВЕДЕНИЕ

Частотно-временные методы анализа звуковых сигналов, основанные на принципах психоакустики, являются базой для построения моделей слухового восприятия. Задача моделирования слухового восприятия возникает во многих приложениях цифровой обработки сигналов. В частности, она возникла в связи с повсеместным распространением цифровых мультимедийных систем. Так, например, перцентуальное (т.е. основанное на модели слухового восприятия) аудиокодирование используется для уменьшения скорости передачи данных в таких алгоритмах, как MP3 и AAC [1]. Перцентуальное аудиокодирование основано на использовании феномена маскирования, который можно рассчитать исходя из того, как сигнал-возбуждение представляется в улитке уха. Системы распознавания речи также используют модель слуха для начального анализа сигнала. Этим эмулируется предварительная обработка сигнала до того этапа, на котором мозг выполняет собственные процедуры распознавания речи. Ценность таких методов анализа звука для систем распознавания речи заключается в их устойчивости к шумам акустической среды [2].

Используемые в настоящее время модели слухового восприятия значительно отличаются по числу моделируемых ими феноменов, присутствующих реальной слуховой системе. Это объясняется тем, что в каждой области применения к модели слуха предъявляются собственные требования. В данной работе дается краткий обзор существующих методов анализа звуковых сигналов, используемых в моделях слухового восприятия.

Методы анализа сигналов, используемые на первом этапе моделирования слухового восприятия, можно условно разделить на *биоинспирированные* и *психоакустически мотивированные*. В биоинспирированных методах ставится задача моделирования работы органов слуховой системы. В частности, при помощи банка цифровых фильтров моделируется механическое движение базилярной мембраны улитки уха. Такие фильтры называют кохлеарными (англ. *cochlear* – улитковый) или слуховыми. Психоакустический мотивированный анализ основывается на закономерностях, выявленных в процессе изучения восприятия слушателем различных акустических стимулов [3].

Существуют несколько биоинспирированных моделей слуховых фильтров:

- roex-фильтры [4];
- гамматон банк фильтров [5];
- модель Кейтса [6];
- гаммачирп банк фильтров [7];

Модели слуховых фильтров получают путем подгонки параметрических выражений под имеющиеся экспериментальные данные. Простое описание слухового фильтра в частотной области можно получить с использованием скругленной экспоненциальной функции (roex-функции):

$$|H(f)|^2 = (1 - r)(1 - pg)e^{-pg} + r,$$

где g – нормализованное отклонение частоты f от центральной частоты фильтра f_c :

$$g = |f - f_c|/f_c.$$

Параметр p характеризует добротность фильтра и должен выбираться, как $p = 4f_c/\text{ERB}(f_c)$, где

$$\text{ERB}(f_c) = 0.1079f_c + 24.7.$$

Второй параметр r определяет пологость формы фильтра за пределом полосы пропускания. Недо-

статком данной модели является то, что слуховые фильтры определяются только в частотной области, а временные параметры фильтров в явном виде не определены.

Гамматон функции [5], используются для описания во временной области импульсных характеристик слуховых фильтров:

$$h(t) = t^{(l-1)} e^{-2\pi b \text{ERB}(f_c)t} \cos(2\pi f_c t), \quad t > 0, \quad (1)$$

где b – параметр, определяющий ширину полосы, а l – порядок. На практике используют значения $l = 4$, $b = 1,019$. Фильтры получили свое название от того, что их огибающая формируется как произведение степенной и экспоненциальной функций, которое является функцией гамма-распределения, а косинусная несущая является тоном, когда её частота находится в слышимом диапазоне.

На практике гамматон фильтры моделируются в виде цифровых КИХ-фильтров. При этом, однако, возникает затруднение от того, что получаемые импульсные характеристики фильтров имеют большую длительность. На рис. 1 приведена импульсная характеристика гамматон фильтра для центральной частоты 500 Гц, где также показана его огибающая. Для сравнения на рис. 1 также приведена огибающая гамматон фильтра с центральной частотой 80 Гц. Можно заметить, что для центральной частоты 500 Гц длина импульсной характеристики фильтра будет составлять порядка 240 отсчетов (при частоте дискретизации 8000 Гц). Для более низких центральных частот, длина импульсной характеристики возрастает ещё больше (например для $f_c = 80$ Гц, она будет содержать 600 отсчетов). Таким образом, получающиеся КИХ-фильтры в вычислительном плане являются довольно затратными.

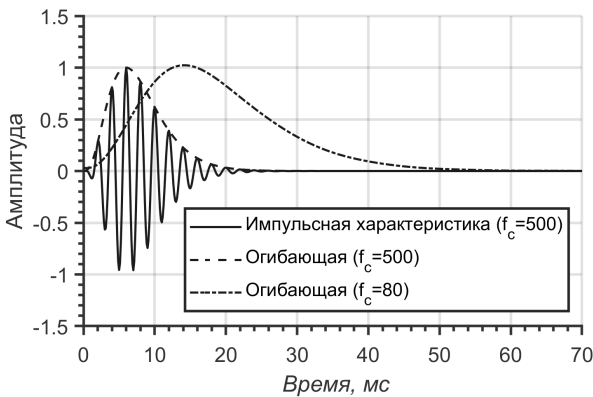


Рис. 1 – Импульсная характеристика и огибающие импульсных характеристик гамматон фильтров для различных центральных частот

В работе [7] была уточнена модель слухового гамматон фильтра, путем добавления

частотно-модулирующего члена:

$$g_c(t) = at^{(l-1)} e^{-2\pi b \text{ERB}(f_r)t} \cos(2\pi f_r t + c \ln t + \phi), \quad t > 0, \quad (2)$$

Единственная разница между выражениями (2) и (1) это наличие члена $c \ln t$, где c – дополнительный параметр. Фильтр имеет монотонно возрастающую частотно-модулированную несущую (чирп-сигнал) с огибающей, которая является функцией гамма-распределения, и отсюда и название «gammachirp». В (2) используется обозначение f_r вместо f_c для параметра частоты, потому что пиковая частота АЧХ более зависит от c и, в меньшей степени, от b и l . Эквивалентная прямоугольная полоса пропускания фильтра зависит от уровня стимула. Для удобства параметр b связывается с уровнем стимула, так что основная формула для ширины фильтра, такая же, как в выражении (1).

Несмотря на то, что гаммачирп фильтры лучше описывают имеющиеся физиологические данные в плане сложности реализации они имеют те же недостатки, что и гамматон фильтры. Более того, гаммачирп фильтры зависят от уровня подаваемого сигнала (в частности изменяется ширина полосы пропускания), что также является препятствием для получения их эффективной реализации.

Алтернативным подходом к построению слуховых фильтров является модель Кейтса, реализующая фильтры бегущей волны [6]. Блок-схема слуховых фильтров Кейтса приведена на рис. 2.

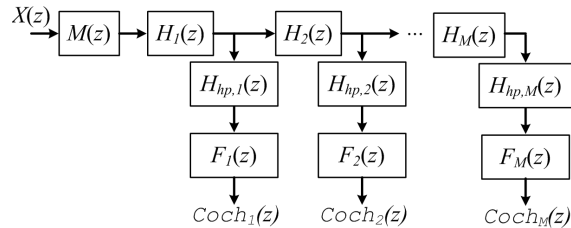


Рис. 2 – Модель слуховых фильтров Кейтса

Важной особенностью этой модели является то, что она представляет собой единую вычислительную структуру. Тем не менее, поскольку вся обработка выполняется во временной области, реализация данной модели также требует значительных вычислительных затрат.

II. ПСИХОАКУСТИЧЕСКИ МОТИВИРОВАННЫЙ АНАЛИЗ

Психоакустика – научное направление, задачей которого является изучение структуры и функции слухового пути с помощью анализа взаимосвязи акустических стимулов, возбуждающих слуховую систему, и психологических (моторных) реакций слушателя на эти системы [3]. В методах психоакустически мотивированного

анализа не ставится задача подогнать какую-либо параметрическую модель под имеющиеся физиологические данные. Целью такого анализа является смоделировать особенности частотной селективности уха. Известно, что слуховая система разбивает весь слышимый частотный диапазон на *критические полосы*. При восприятии звуков со сложным спектром, громкость частотных компонент, попадающие в одну критическую полосу, суммируется. Таким образом, компоненты с различной частотой, в том случае, когда они попадают в одну критическую полосу, воспринимаются, как единый сигнал.

Можно выделить следующие три группы психоакустически мотивированных методов:

- ДПФ с суммированием энергий в субполосах;
- банки фильтров на основе вейвлет преобразования [8];
- банки фильтров на основе фазового преобразования [9];

Согласно первому из перечисленных методов, неравномерное спектральное разрешение слуховой системы учитывается путем суммирования энергий соответствующего числа соседних частотных субполос ДПФ. Следовательно, фазовое соотношение между спектральными компонентами различных субполос в пределах полосы слухового фильтра не принимается во внимание. Поскольку фронты слухового фильтра менее крутые, чем крутизна субполос, они должны быть аппроксимированы путем распределения энергий субполос по нескольким критическим полосам. Данный способ отображения равных субполос в полосы слуховых фильтра дает огибающие выходного сигнала, которые отличаются от тех, которые измеряются на выходе слуховой улитки. Временное разрешение такого спектрального разложения определяется размером преобразования (т.е. длиной ДПФ), и, таким образом, является постоянным для всех центральных частот. Для высокочастотных полос это приводит к значительно более низкому временному разрешению по сравнению с соответствующими слуховыми фильтрами. Все описанные несоответствия способствуют неточному моделированию частотного анализа, производимого ухом.

Банки фильтров, получающиеся на основе вейвлет преобразования имеют большое значение для моделирования частотной селективности уха. В частности, при использовании вейвлет банков фильтров достигается разрешение с постоянной добротностью (англ. *constant-Q resolution*) [10], при котором временное разрешение возрастает, а частотное уменьшается по мере увеличения частоты. В отличие от предыдущего метода анализа, банки фильтров на основе вейвлетов имеют неравномерное временное разрешение, что отражает свойство реальной слуховой

системы. Тем не менее, сложность вейвлет банков фильтров заключается в том, что производимая ими частотно-временная декомпозиция сигнала напрямую зависит от древовидной структуры банка фильтров. Также ширина критических полос не достаточно точно моделируется при помощи таких банков фильтров [11-12].

Банки фильтров, основанные на применении фазового преобразования, являются альтернативным методом психоакустически мотивированного анализа по отношению к вейвлет банкам фильтров [13]. Особенностью данного вида банков фильтров является то, что они образуются из обычных равнополосных структур анализа путем замены элементов задержки на фазовые звенья:

$$z^{-1} \rightarrow A(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \quad A(e^{j\omega}) = e^{j\theta(\omega)}, \quad (3)$$

где α – параметр отвечающий за степень растяжения частотной оси, в результате которой банк фильтров получает свойство неравнополосности. Выражение (3) носит название *фазового преобразования* [14]. В результате применения (3) к линейной дискретной системе происходит трансформация частотной оси, описываемая выражением:

$$\theta^{-1} : \Omega \rightarrow \omega, \quad (4)$$

где Ω – деформированная частотная ось, ω – исходная частотная ось, а θ – функция, отображающая ω в Ω :

$$\theta(\omega) = \omega + \arctg\left(\frac{\alpha \sin \omega}{1 - \alpha \cos \omega}\right). \quad (5)$$

Таким образом, при использовании фазового преобразования исходный банк фильтров является равнополосным относительно деформированной частотной оси Ω . Однако, относительно частоты ω на которой производится обработка сигнала в силу (4) банк фильтров является неравнополосным. Описанный процесс иллюстрируется рис. 3, где показано получение 8-канального неравнополосного ДПФ-модулированного банка фильтров. Более подробно механизм фазового преобразования описан в [13].

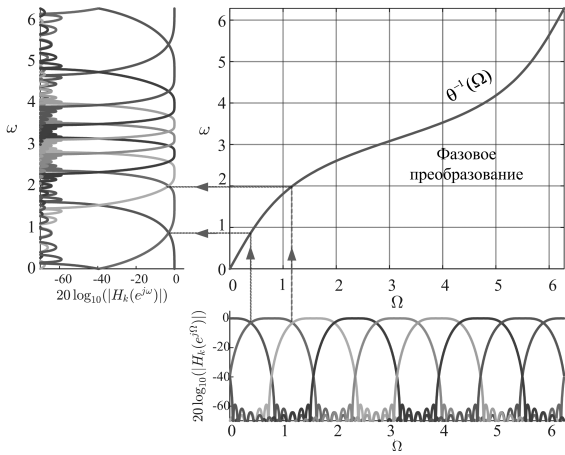


Рис. 3 – Процесс получения неравнополосного банка фильтров с использованием фазового преобразования, коэффициент определяющий степень деформации частотной оси $\alpha = 0,4$ [13]

Существенным отличием банков фильтров на основе фазового преобразования является то, что выполняемая ими частотно-временная декомпозиция сигнала не имеет жесткой привязки к структуре банка фильтров (как это имеет место для вейвлет банков фильтров). Также использование фазового преобразования позволяет более точно аппроксимировать психоакустическую шкалу барков [11, 15], чем банк фильтров на основе вейвлетов. К недостаткам банков фильтров на основе фазового преобразования можно отнести то, что их структура включает цепочку фазовых звеньев, которая может содержать более 100 элементов [9,11]. Вычисления производимые в цепочке фазовых звеньев трудно распараллелить, и как следствие при её программной реализации тратится значительное процессорное время. При аппаратной реализации цепочки фазовых звеньев такой проблемы не возникает [16].

III. Выводы

В докладе рассмотрены основные методы биоинспирированного и психоакустически мотивированного анализа звуковых сигналов, используемых в современных информационных системах и технологиях. Изначально применение описанных методов анализа было обусловлено бурным развитием систем кодирования речи и аудио в 90-х года прошлого столетия [1], далее их стали активно применять в системах распознавания речи [2]. На настоящем этапе биоинспирированные и психоакустически мотивированные методы анализа звуков применяются в различных информационных системах, осуществляющих шумоподавление [11-12], конверсию голосового сигнала, повышение разборчивости речи. Особо можно выделить медицинские системы, в которых также используются указанные методы, это слуховые аппараты [16], акустические анализаторы голоса, программные комплексы для выявления патологии по голосу [17].

СПИСОК ЛИТЕРАТУРЫ

1. Ковалгин, Ю. А. Аудиотехника / Ю. А. Ковалгин, Володин Э. И. – М.: Горячая линия–Телеком, 2013. – 742 с.
2. Kim, D. S. Auditory processing of speech signals for robust speech recognition in real-world noisy environments / D. S. Kim, S. Y. Lee, R. M. Kil // IEEE Transactions on speech and audio processing. – 1999. – vol. 7, №. 1. – P. 55–69.
3. Слуховая система / ред. Я. А. Альтман. – Л.: Наука, 1990. – 620 с.
4. The deterioration of hearing with age: Frequency selectivity, the critical ratio, the audiogram, and speech threshold / R. D. Patterson, *et al.* // The Journal of the Acoustical Society of America. – 1982. – vol. 72, №. 6. – P. 1788–1803.
5. An efficient auditory filterbank based on the gammatone function / R. D. Patterson, *et al.* // Proceeding of the IOC Speech Group on Auditory Modelling at RSRE. – 1987. – vol. 2, №. 7. – P. 1-34.
6. Kates, J. M. A time-domain digital cochlear model // IEEE Transactions on Signal Processing. – 1991. – vol. 39, №. 12. – P. 2573–2592.
7. Irino, T. A time-domain, level-dependent auditory filter: The gammachirp / T. Irino, R. D. Patterson // The Journal of the Acoustical Society of America. – 1997. – vol. 101, №. 1. – P. 412-419.
8. Петровский, Ал. А. Масштабируемые аудиоречевые кодеры на основе адаптивного частотно-временного анализа звуковых сигналов / Ал. А. Петровский, Петровский А. А. // Труды СПИИРАН. – 2016. – Т. 1, №. 50. – С. 55–92.
9. Вашкевич, М. И. Проектирование передискретизированного неравнополосного косинусно-модулированного банка фильтров / М. И. Вашкевич, А. А. Петровский // Информатика. – 2011. – №. 2. – С. 21–39.
10. Quatieri, T.F. Discrete-time speech signal processing: principles and practice / Prentice Hall Signal Processing Series. — Prentice Hall PTR, 2002. — P. 781.
11. Gulzow, T. Comparison of a discrete wavelet transformation and a nonuniform polyphase filterbank applied to spectral-subtraction speech enhancement / T. Gulzow, A. Engelsberg, U. Heute // Signal processing. – 1998. – Vol. 64, no. 1. – pp. 5–19.
12. Carnero, B. Perceptual speech coding and enhancement using frame-synchronized fast wavelet packet transform algorithms / B. Carnero, A. Drygajlo // IEEE Transactions on Signal Processing. – 1999. – vol. 47, №. 6. – P. 1622–1635.
13. Вашкевич, М.И. Сравнение частотно-временных преобразований: Фурье-анализ, вейвлеты и банки фильтров на основе фазового преобразования / М. И. Вашкевич, И. С. Азаров // Цифровая обработка сигналов – 2020. – №2. – С. 13–26.
14. Vary, P. Digital filter banks with unequal resolution // Short Communication Digest of European Signal Processing Conference (EUSIPCO), 1980. – pp. 41–42.
15. Smith, J. O. Bark and ERB bilinear transforms / J. O. Smith, J. S. Abel // IEEE Transactions on speech and Audio Processing. – 1999. – vol. 7, №. 6. – P. 697–708.
16. Вашкевич, М. И. Неравнополосные банки фильтров для слуховых аппаратов: анализ алгоритмов, автоматизация проектирования / М. И. Вашкевич, А. А. Петровский // Автоматизация проектирования дискретных систем: материалы 7-й междунар. конф. Беларусь, Минск, 16–17 ноября. — 2010. – С. 53-60.
17. Вашкевич М. И., Азаров И. С. Определение патологии голосового аппарата на основе анализа модуляционного спектра речи в критических полосах / М. И. Вашкевич, И. С. Азаров // Труды СПИИРАН. – 2020. – Т. 19. – №. 2. – С. 249–276.

МЕТОДЫ И СРЕДСТВА СИСТЕМЫ ГЕОБАЗАДАННЫХ ДЛЯ АДАПТАЦИИ КОМПЬЮТЕРНЫХ МОДЕЛЕЙ. ПРИМЕРЫ АДАПТАЦИИ

Таранчук В. Б.

Кафедра компьютерных технологий и систем, Факультет прикладной математики и информатики,
Белорусский государственный университет
Минск, Республика Беларусь
E-mail: taranchuk@bsu.by

Обсуждаются способы интерактивного формирования в вычислительных экспериментах отвечающих интуитивным требованиям эксперта цифровых моделей геологических, геоэкологических объектов. Отмечены специальные инструменты системы ГеоБазаДанных, результаты сопоставления с эталонными решениями, получаемыми в комплексе Генератор Геологической Модели Залежи. Представительными примерами иллюстрируется эффективность алгоритмов аппроксимации и восстановления цифрового поля, его интерактивной адаптации в системе ГеоБазаДанных.

ВВЕДЕНИЕ

В первой части данной работы отмечены актуальность и общие подходы к решению задач разработки и внедрения различных компьютерных геологических моделей с самонастраивающимися средствами их адаптации. Ключевыми вопросами являются автоматизация, адаптация моделей с учетом непрерывно поступающих геоданных, а также пересмотр и уточнение результатов обработки информации с использованием новых методов интерпретации. Несколько способов, методические решения и программные средства системы ГеоБазаДанных, позволяющие подтвердить достоверность интерпретаций, визуализировать и получить численные значения погрешностей рассчитанных различными методами интеллектуальной обработки результатов, включаемых и используемых в компьютерных геологических, геоэкологических моделях, рассматриваются ниже. Выбрана ключевая задача формирования и обработки используемых в компьютерных моделях цифровых полей – обсуждаются предложенные методы, реализованные в интерактивной компьютерной системе ГеоБазаДанных специализированные алгоритмы аппроксимации, которые апробированы для решения различных прикладных задач.

I. ПРИМЕРЫ ВОССТАНОВЛЕНИЯ ЦИФРОВЫХ ПОЛЕЙ

Приведем несколько нестандартных решений, которые признаны сложными для всех пакетов обработки геоданных. Задача реконструкции цифрового поля предполагает вычисления значений аппроксимирующей функции в узлах регулярной сетки по значениям хаотически расположенных экспериментальных точек данных (наблюдений), т.е. – создание регулярного массива Z -значений узловых точек регулярной сетки X, Y по нерегулярному массиву (x, y, z) -значений. Термин «нерегулярный массив значений» означает,

что x, y координаты точек данных распределены по области определения функции неравномерно.

Процедура построения регулярной сетки, восстановления сеточной функции значений уровней моделируемой поверхности представляет собой интерполяцию или экстраполяцию по набору рассеянного множества исходных точек и значений уровней поверхности в них на равномерно распределенные узлы в исследуемой области (на площади). Методы восстановления сеточных функций и соответствующие алгоритмы реализованы в нескольких специализированных пакетах компьютерной графики, ГИС ([1]). Следует понимать, что восстановление сеточной функции не предполагают нахождение единственного решения некоторой математической задачи. Субъективное мнение, квалификация эксперта – факторы, всегда присутствующие в подобной деятельности. Составителю компьютерной модели важно иметь инструменты, позволяющие «поиграть» с исходными данными и сопоставить результаты с подготовленными эталонами, выполнять интерактивно обработку данных; иметь средства имитации возможных ситуаций получения и коррекции входной информации, модули математической обработки и статистического анализа.

Приведенные ниже примеры работы в системе ГеоБазаДанных, в частности, оценки точности результатов с использованием инструментов программного комплекса Генератор Геологической Модели Залежи (ГГМЗ) иллюстрируют уникальные возможности методов и программного обеспечения системы ГеоБазаДанных. Разработанные алгоритмы интерактивной подготовки и адаптации цифровых распределений иллюстрируются представительными примерами с использованием исходных данных работ [2], [3]. При этом для формирования эталонного распределения (поверхности) в качестве возмущений не рассматриваются типичные графические объекты (пирамиды, параллелепипеды, цилиндры,

конусы). Причина в том, что в системе ГеоБазаДанных трехмерные геометрические шаблоны могут обрабатываться отдельно, воспроизводиться точно. Специальный модуль системы может быть использован для «восстановления поверхностей, когда цифровое поле не вычисляется методом аппроксимации на выбранных участках, а заполняется вычисленными значениями функций из библиотеки шаблонов» [1]. Также для демонстрации особых возможностей системы ГеоБазаДанных, помимо возмущений типа «гладкие формы», рассмотренных в [2] и [3], в эталоне вводятся возмущения, имитирующие плато и разрыв. Они задаются выражениями, приведенными в [1]. Используемая эталонная поверхность формируется в ГГМЗ. Эксперт конструирует поверхность интерактивно путем визуального размещения возмущений на опорной поверхности-ленте (регламент действий изложен в [1 – 3]). Дополнительно отметим, что эталонная поверхность имеет точное математическое описание. Рассматриваемую в примерах эталонную поверхность и ограничиваемый ей объем иллюстрирует Рис. 1.

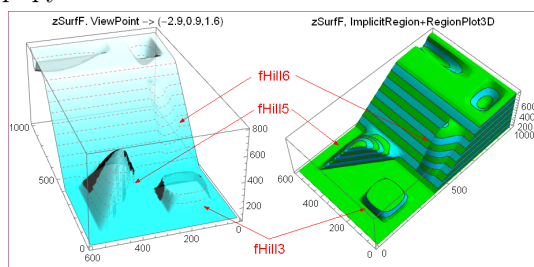


Рис. 1 – Графики поверхности и объема zSurfF

Акцентируем внимание на фрагментах поверхности zSurfF, а именно, возмущениях fHill3, fHill5, fHill6. Элемент fHill3(200,150) имитирует наличие плато (отсечение горизонтальной плоскостью), fHill5 (550,150) и fHill6 (250,400) имитируют разрыв (в геологии – сдвиг-надвиг), математически описывается отсечением части гладкой формы типа холм вертикальной плоскостью (в идентификаторах в скобках даны координаты центров возмущений). Подобные фрагменты fHill3, fHill5, fHill6 средствами восстановления цифровых полей стандартными методами не воспроизводятся. Также отметим, что в следующих иллюстрациях изолинии соответствующие уровням плато (для zSurfF – уровень 225) не выводятся. На иллюстрации zSurfF уровень 210 показан, чтобы дополнительно акцентировать внимание на границах возмущений, размещенных на участке плато.

На Рис. 2, 3 приведены визуализации примеров реконструкции сеточных функций с использованием стандартных методов, алгоритмов, а также уникальных методов «подгонки» распределений в системе ГеоБазаДанных. Предварительно были выполнены типовые этапы имитации наблюдений и подготовки исход-

ных данных ([1]) для последующей реконструкции цифрового поля из рассеянного набора точек с измерениями. Предполагается, что пункты замеров уровней расположены на профилях; они показаны для разных профилей наблюдения разными примитивами (112 точек). Центры форм возмущений (фрагментов-возмущений) отмечены красными кружками, их 6 точек. Две пары красных маркеров ромбиков показывают направления разрывов. В работе не было цели подобрать хорошую систему профилей, таковые сформированы эскизно, примерно, как часто подобное делают проектировщики.

Рис. 2 иллюстрирует результаты, полученные в комплексе ГГМЗ и генераторе Gen_MAPw. Восстановленное в ГГМЗ поле рассчитано методом интерполяции 1-го порядка в Wolfram Mathematica. Соответствующие иллюстрации показаны на контурной карте, где кроме профилей и пунктов замеров выведены изолинии: эталонного поля – сплошными фиолетовыми линиями, в восстановленном поле – пунктирными черными линиями.

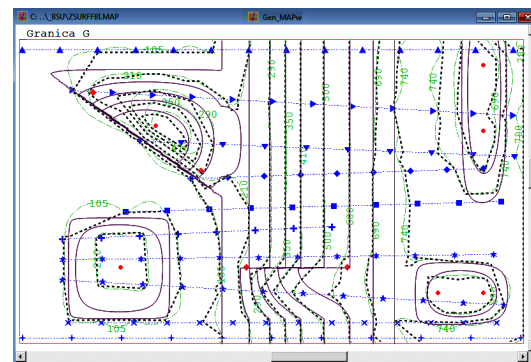


Рис. 2 – Изолинии цифровых полей эталонного и восстановленных в ГГМЗ и Gen_MAPw

Для получения результатов в системе ГеоБазаДанных значения (x,y,z) эталонной функции в контрольных точках из ГГМЗ экспортируются в XLS-файл, который импортируется в Gen_MAPw. На начальном этапе для сравнения результатов восстановления цифрового поля расчеты реконструкции поля методом сплайн-аппроксимации (версии системы ГеоБазаДанных) проводились в Gen_MAPw без каких-либо дополнений – соответствующие изолинии отображаются штрих-пунктирными зелеными полилиниями. Расчетная сетка была сформирована в Gen_MAPw автоматически с учетом покрытия всей площади, окаймленной границей Granica G – в этом случае работает элемент системы «Подмножество». На иллюстрации существует понятная разница в изолиниях, поскольку результаты получаются с использованием различных методов интерполяции. В основном оба метода обеспечивают визуализацию форм возмущений fHill, но со значительными ошибками воспроизведения для fHill5 и fHill6. Результаты Рис. 2 (применения типовых математических методов)

не стоит комментировать, потому что в практике создания компьютерных моделей они именно такими часто и бывают, хотя, на самом деле, не пригодны для использования.

II. ПРИМЕРЫ ИНТЕРАКТИВНОЙ АДАПТАЦИИ ЦИФРОВЫХ ПОЛЕЙ

Как понимать результаты, показанные на Рис. 3? В приводимом здесь случае и подобных задача состоит в корректировке цифровых полей, полученных алгоритмом аппроксимации, с помощью таких ГеоБазаДанных элементов, как «Выделено», «Разрыв» и «Граничные условия» [4]. Система ГеоБазаДанных «распознает» и соответствующими способами интерпретирует специальные контуры (в частности, полилинии категорий: подобласти, включения, граничные условия, разрывы) по атрибутам линий, к которым относятся цвет, толщина, тип линии, замкнутая или нет, способ заполнения в случае замкнутого контура. По сути, на Рис. 3 иллюстрируется интерактивная интеллектуальная адаптация. Поясним результаты.

Первоначально про элемент типа «Разрыв». Разрыв – дополняемый во входную карту незамкнутый контур (ломаная) без самопересечений с специально для него назначенными пользователем атрибутами. В системе ГеоБазаДанных при наличии во входной карте разрывов вблизи их расчеты цифрового поля, изолиний, зон относительной однородности выполняются специальным образом. В частности, в алгоритме аппроксимации обеспечивается такой эффект наличия контура-разрыва, что опорные точки по одну сторону от такого контура не оказывают влияния на сеточные значения по другую сторону. При построении тематических геологических карт разрывами моделируются тектонические разломы, сдвиг/надвиг пластов.

Выполнение аппроксимации (формирование цифрового поля на сетке) в варианте Рис. 3 – во входную карту были добавлены 2 контура – жирные штрих-пунктирные красные линии (другие дополнения – пояснения далее). Эффект дополнения элемента «Разрыв» выразительно демонстрируется в левой и нижней центральной части, особенно, для изолиний 290, 350, 410, 500, 600.

Также на Рис. 3 иллюстрируются возможности дополнительных уточнений цифрового поля, которые должны соответствовать априорному мнению пользователя с использованием инструментов «Выделено» и «Везде». Участки «Выделено» обеспечивают автономный расчет в узлах внутри ограниченных соответствующим контуром подобластей. Принадлежащие этим участкам точки с замерами вне участков «Выделено» не учитываются.

Изолинии в версии расчета с добавлением к входным данным перечисленных элементов вы-

ведены зелеными полилиниями. Эффекты повышения точности дополнением элементов адаптации на Рис. 3 наглядны.

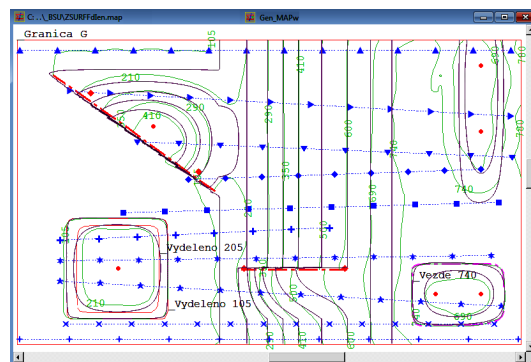


Рис. 3 – Изолинии цифровых полей эталонного и восстановленного в Gen_MAPw в варианте с адаптацией

Следует отметить, что первоначальные возможности системы ГеоБазаДанных существенно расширены за последние годы. Это достигнуто интеграцией и дополнением системы инструментами комплекса ГГМЗ, доступом к функциям системы компьютерной алгебры. В текущем состоянии ГеоБазаДанных обеспечивает пользователей не только средствами решения конкретных производственных задач, но и возможностями научного исследования новых методов анализа и обработки исходных данных и применяемых компьютерных моделей. В частности, для приведенной выше задачи получены оценки погрешностей по методике, изложенной в [1, 3]. Здесь конкретные цифры не приводятся, потому что погрешности для рассчитанного адаптированного поля сопоставимы с точностью получения сеточных функций, и эти погрешности на порядок ниже, чем для неадаптированного поля.

ЗАКЛЮЧЕНИЕ

Представлены результаты использования инструментов системы ГеоБазаДанных при интерактивной адаптации цифровых полей компьютерных моделей.

СПИСОК ЛИТЕРАТУРЫ

1. Таранчук, В. Б. Компьютерные модели подземной гидродинамики / В. Б. Таранчук // Минск : БГУ, 2020. – 235 с.
2. Таранчук, В. Б. Интегрированный программный комплекс тестировщика геологических моделей / В. Б. Таранчук // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. –2017. –№ 16 (265). –С. 148–159.
3. Таранчук, В. Б. Методы и инструментарий оценки точности компьютерных геологических моделей / В. Б. Таранчук // Вестник БрГТУ, № 2, –2018. – С. 64–70.
4. Таранчук, В. Б. Построение цифровых геолого-экологических моделей в системе ГБД-э / В. Б. Таранчук // Геоэкология Беларуси – Минск : Право и экономика, Научное издание, 2006. – С. 72–96.

Секция «Системы управления»

Section «Control systems»

УПРАВЛЕНИЕ КРАТКОСРОЧНЫМИ КОЛЕБАНИЯМИ И ФЛУКТУАЦИЕЙ В РАЗВИТИИ ОСНОВНЫХ ФОНДОВ ОТРАСЛИ ЭКОНОМИЧЕСКОЙ СИСТЕМЫ

Бейсенби М. А., Бейсембина С. Е.
Евразийский Национальный Университет имени Л.Н. Гумилева
г. Нур-Султан, Республика Казахстан
E-mail: beisenbi@mail.ru, sshamshiyeva@gmail.com

Развитие экономических процессов рассматривается, как нелинейная динамическая система и концепция детерминированного хаоса является актуальным и объясняет порождение и развитие краткосрочных колебаний и флуктуаций в экономической системе. Предлагается метод управления краткосрочным колебанием и флуктуацией в классе однопараметрических структурно устойчивых отображений на базе нелинейной динамической модели развитие основных фондов отрасли экономической системы. Условие отсутствия краткосрочных колебаний и флуктуаций получено в форме простейших неравенств относительно обобщенных параметров экономической системы, характеризующие основное свойство динамической системы – устойчивость.

ВВЕДЕНИЕ

Проблема детерминированного хаоса является одним из фундаментальных концепций в развитии теорий современных динамических систем. Исследований конца XX-века выявили большое разнообразие динамики нелинейных систем и привели к одному из важнейших открытий в нелинейных динамических системах – детерминированному хаосу с порождением «странного аттрактора» [1, 2, 3]. Детерминированный хаос в основном оказывает вредные воздействия на систему и могут проявляться в технологических системах в форме «разноса» и «аварий», в социально-экономических, биологических, медицинских и т.д. системах в форме «кризиса». Режим детерминированного хаоса в нелинейной, динамической экономической системе проявляется как краткосрочные колебаний и флуктуации [4,5], которые при определенной частоте и амплитуде хаотических колебаний, вызывает экономической системе «кризис», так как бизнес сворачивает экономическую активность [6]. Хаотические режимы действительно могут возникнуть в любой системе. Поэтому возникли практически важные классы задач, когда нелинейной системой необходимо управлять, уменьшая или наоборот увеличивая степень ее хаотичности. Существующие методы: стабилизация неустойчивой периодической орбиты задачи хаотизаций, задачи управляемой синхронизации и модификаций аттракторов [7,8,9,10], не решают проблему подавления или исключения из режима динамической системы детерминированного хаоса [4,5,11,12]. В настоящей работе предлагается метод построения системы управления краткосрочными колебаниями и флуктуацией, на примере нелинейной динамической модели развитие основных фондов одной отрасли в многоотраслевой экономической системе. Предлагается новый подход к управлению хаотическими процессами в классе однопараметрических структурно-

устойчивых отображений из теорий катастроф [13,14]. Где путем выбора закона управления правая часть нелинейного динамического управления развитие основных фондов отрасли экономической системы представляется в форме однопараметрических структурно-устойчивых отображений [13,14]. Условия отсутствия краткосрочных колебаний и флуктуаций в системе исследуется градиентно-скоростным методом вектор функций Ляпунова [12,15,16,17]. Динамику основных фондов отдельной отрасли экономической системы можем привести к упрощенной модели [18]:

$$\frac{dx}{dt} = -\frac{\gamma_1}{T}x^2 + \frac{\alpha}{T}x, \quad (2)$$

Закон управления инвестиций $u(t)$ в основные фонды выбираем таким образом, чтобы правая часть уравнения состояния (2) имела форму однопараметрических структурно устойчивых отображений [14,15]

$$u(t) = -x^3 + \frac{\gamma_1}{T}x^2 + kx \quad (3)$$

Уравнение динамики развитие основных фондов отрасли экономической с учетом управления (3) получим в виде

$$\frac{dx}{dt} = -x^3 + \left(\frac{\alpha}{T} + k\right)x \quad (4)$$

Рассмотрим равновесные состояния системы (4):

$$-x_s^3 + \left(\frac{\alpha}{T} + k\right)x_s = 0 \quad (5)$$

Тривиальное решение уравнения (5)

$$x_s^1 = 0 \quad (6)$$

и нетривиальное решение, определяемое решением уравнения

$$-x_s^2 + \left(\frac{\alpha}{T} + k\right) = 0$$

При отрицательном $\frac{\alpha}{T} + k \leq 0$ это уравнение имеет мнимое решение, что не может соответствовать какой-либо экономической или физически возможной ситуации. Однако при положительных $\frac{\alpha}{T} + k > 0$ это уравнение допускает следующие два решения:

$$x_s^{2,3} = \pm \sqrt{\frac{\alpha}{T} + k}, \quad (7)$$

Эти решения сливаются с x_s^1 при $\frac{\alpha}{T} + k = 0$ и ответвляются от него при $\frac{\alpha}{T} + k > 0$. Это так называемое явление бифуркации. Несмотря на свою кажущуюся простоту, нелинейная динамическая модель (2) поражает невообразимым многообразием типов поведения, варьирующего от простых точек равновесия до множественных периодических или хаотических [4,5] в зависимости от значения обобщенного параметра $\frac{\alpha}{T}$. Основные фонды развиваются без колебаний до тех пор пока выполняется условия $\alpha \leq 2T$. При изменении $\frac{\alpha}{T}$ в интервале $(2T < \alpha < T * 2,5699 \dots)$ происходит бесконечная последовательность бифуркаций, каждое из которых приводит к циклам более высокого порядка с периодом, удваивающимся при каждой последовательной бифуркации. После $(T * 2,5699 \dots) < \alpha$ получается орбиты с «бесконечным периодом», т.е. с ярко выраженным хаотическим аттрактором, характеризующему неустойчивостью. Таким образом краткосрочные колебания и флуктуации по модели развитие основных фондов отрасли (2) экономической системы порождается в условиях потери устойчивости. Поэтому покажем, что предложенный подход к управлению развитием основных фондов позволяет построить систему управления (4), устойчивая при любом изменении неопределённых параметров. Режим краткосрочных колебаний и флуктуаций в развитии основных фондов отрасли по модели (4) характеризуется границами аperiodической устойчивости, где в переходном процессе отсутствует периодические и хаотические колебаний. Следовательно, исследуется устойчивость стационарных состояний (6) и (7) градиентно-скоростным методом вектор функции Ляпунова [12,15,16]. Сначала исследуем устойчивость стационарного состояния (6). Из (5) определяем градиента функции Ляпунова $V(x)$:

$$\frac{\partial V(x)}{\partial x} = x^3 - \left(\frac{\alpha}{T} + k\right) x, \quad (8)$$

Также из (5) определяем вектор скорости

$$\frac{dx}{dt} = -\left(x^3 - \left(\frac{\alpha}{T} + k\right) x\right) \quad (9)$$

Полная производная по времени от функции Ляпунова определяется, как скалярное произведение вектора градиента (8) на вектор скорости (9):

$$\frac{dV(x)}{dt} = -\left(x^3 - \left(\frac{\alpha}{T} + k\right) x\right)^2 \quad (10)$$

Из (9) очевидно, что полная производная от функции Ляпунова является знакоотрицательной функцией, то есть достаточное условие асимптотической устойчивости системы (4) гарантированно выполняется. По градиенту (9) построим функции Ляпунова

$$V(x) = \frac{1}{4}x^4 - \frac{1}{2}\left(\frac{\alpha}{T} + k\right)x^2 \quad (11)$$

Условия положительной определенности функций Ляпунова (11) на стационарном состоянии (6):

$$\frac{\alpha}{T} + k < 0 \quad (12)$$

Отсюда стационарное состояние (6) системы (4) существует и является асимптотической устойчивой, если выполняется условия (12). Исследуем устойчивость стационарного состояния (7). Для этого уравнения состояния (4) представим в отклонениях относительно стационарного состояния (7):

$$\frac{dx}{dt} = -x^3 - 3\sqrt{\frac{\alpha}{T} + k}x^2 - 2\left(\frac{\alpha}{T} + k\right)x, \quad (13)$$

Из (13) определяем градиент функции Ляпунова $V(x)$:

$$\frac{\partial V(x)}{\partial x} = x^3 + 3\sqrt{\frac{\alpha}{T} + k}x^2 + 2\left(\frac{\alpha}{T} + k\right)x \quad (14)$$

Из (13) вектор скорости системы имеет вид

$$\frac{dx}{dt} = -x^3 - 3\sqrt{\frac{\alpha}{T} + k}x^2 - 2\left(\frac{\alpha}{T} + k\right)x \quad (15)$$

Полная производная от функции Ляпунова определяется, как скалярное произведение вектора градиента (14) на вектор скорости (15)

$$\frac{dV(x)}{dt} = -\left[x^3 + 3\sqrt{\frac{\alpha}{T} + k}x^2 + 2\left(\frac{\alpha}{T} + k\right)x\right]^2 \quad (16)$$

Из (16) очевидно, что полная производная от функции Ляпунова является знакоотрицательной функцией. По градиенту (14) строим функции Ляпунова:

$$V(x) = \frac{1}{4}x^4 + \sqrt{\frac{\alpha}{T} + k}x^3 + \left(\frac{\alpha}{T} + k\right)x^2 \quad (17)$$

Условия положительной определенности функций (17) неочевидно, поэтому воспользуемся леммой Морса из теорий катастроф [14,15]. По лемме Морса, функцию Ляпунова (17) локально в окрестности стационарного состояния (7) можем представить в виде квадратичной формы:

$$V(x) = \left(\frac{\alpha}{T} + k\right)x^2 \quad (18)$$

Отсюда условия существования функций Ляпунова, то есть положительная определенность

функций Ляпунова (18) или (17) будет определяться

$$\frac{\alpha}{T} + k > 0 \quad (19)$$

Таким образом система (4) за счёт введения в контур, закон управления в форме однопараметрических структурно устойчивых отображений становится устойчивой в широких пределах изменения параметров α , T и k . Стационарное состояние (16) системы (4) существует и устойчивый при выполнении условий (12), а стационарное состояние (7) существует и устойчивый при выполнении условий (19). Это показывает отсутствие краткосрочных колебаний и флуктуаций в режимах развитие основных фондов отрасли и экономической системы в целом.

Заключение. В настоящее время хаотичности динамики была признано во всех нелинейных динамических системах в механике, системах телекоммуникации, во всех областях физики, химии и биохимии, биологии, в экономике, медицине и т.д. Решаются практически важные классы задач: стабилизация периодической орбиты системы, возбуждения, или генерация хаотических колебаний, управляемой синхронизаций и модификация аттракторов. Но задача подавления или исключения из режима функционирования системы колебательных или хаотических процессов остается нерешенным. Предлагается метод построения системы управления по математической модели развитие основных фондов отрасли экономической системы. Выбор закона управления осуществляется в классе однопараметрических структурно-устойчивых отображений, т.е. правая часть уравнений состояния приводится к форме катастроф складки. Режим краткосрочных колебаний и флуктуаций порождается в системе при потере устойчивости стационарного состояния. Устойчивость стационарных состояний исследуется градиентно-скоростным методом вектор функции Ляпунова. Показано, что краткосрочные колебаний и флуктуаций будут отсутствовать при любых изменениях параметров и это исключает порождение краткосрочных колебаний и флуктуации в системе.

СПИСОК ЛИТЕРАТУРЫ

1. Lorenz E.N. Deterministic nonperiodic flow // J. Atmospheric Science. 1963 Vol.20. №2, P.130-141
2. Ruelle D., Tacens F. On the nature of turbulence // Comm. Math. Physics. 1971. Vol.20. №2, P. 167-192 (Перев. В кн. «Странные аттракторы». М.: Мир, 1981. С.116-151).
3. Брок У. Теория хаоса. – М.: Наука, 2001. – 424 с.
4. Бейсенби М.А., Ойнаров Р.О., Ойнаров А.Р. Экономические флуктуации в краткосрочном периоде и детерминированный хаос. – Известия НАН РК. Серия физико-математическая. – 2005. – №3. – С.30-37.
5. Бейсенби М.А., Ойнаров А.Р. Сценарии возникновения краткосрочных колебаний и флуктуаций на рынке товаров // Доклады НАН РК. – 2005. – № 6. – С.5-12.
6. Грегори Мэнкью Н. Принципы экономикс – СПб: Питер, 2002. – 496 с.
7. Лоскутов А.Ю. Хаос и управление динамическими системами / в кн.: Нелинейная динамика и управление. Под ред. С.В. Емельянова и С.К. Коровина. – Т.1. – М.: Физмат-лит., 2001. – С.163-216.
8. Grebogi C., Lai Y.C., Hayes S. Control and applications of chaos // J. Franklin Inst. 1997. V.334B. P.1115-1146.
9. Grebogi C., Lai Y.C. Controlling chaos in high dimensions // IEEE Trans. Circ. Syst. – I. 1997. V.44. P.971-975.
10. Grebogi C., Lai Y.C. Controlling chaotic dynamical systems // Syst. Contr. Lett. 1997. V.31. №3. P.307-312.
11. Бейсенби М.А. Модели и методы системного анализа и управление детерминированным хаосом в экономике. Астана, 2011. – 201 с.
12. Бейсенби М.А. Исследование робастной устойчивости систем автоматического управления методом функции А.М. Ляпунова. – Астана: DR-Project, 2015. – 204 с.
13. Гилмор Р. Прикладная теория катастроф: В 2-х книгах. Кн. 1 – М.: Мир, 1984. – 350 с.
14. Постон Т., Стюарт И. Теория катастроф и ее приложения. – Мир, 1980. – 607с.
15. Beisenbi M.A., Uskenbayeva G., Satybalдина D. Research of robust stability of control system using a new approach to the Lyapunov functions construction // Modern applied science. – V.9. - 10. -2015. – P.1-16
16. Beisenbi M.A., Uskenbayeva G., Kaliyeva S. Construction and investigation aircraft control system in a class of one-parametric structurally stable mapping using Lyapunov functions. Electronics, communications and networks. Published by Taylor and Francis Group, 2015. –P. 683-689.
17. Beisenbi M.A., Shukirova A.K. Design of control system with increased potential of robust stability for nonlinear object using Lyapunov function. “Technics. Technology. Education. Safety”. Proceedings of IV International scientific technical conference. – Bulgaria: Scientific Technical Union of Mechanical Engineering, 2016. -V.2.-P.38-41.
18. Бейсенби М.А. Управляемый хаос в развитии экономической системы: Монография. –г.Нур-Султан: ТОО «Мастер По», 2019. -168с.

ЦИФРОВЫЕ ТЕХНОЛОГИИ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Акинфина М. А.

Кафедра информационных технологий, Белорусский государственный экономический университет
Минск, Республика Беларусь
E-mail: {akinфина}@rambler.ru

В данной работе приводятся основные элементы внедрения цифровых технологий в деятельность предприятия, которые способствуют появлению новых возможностей оптимизации бизнеса.

ВВЕДЕНИЕ

В последние годы промышленное производство по всему миру претерпевает значительные изменения. Распространение индустриального интернета вещей, «больших данных», роботизации и других технологий подталкивают нас к очередной промышленной революции. Предприятиям и организациям предстоит измениться, чтобы приспособиться к новым реалиям цивилизации.

I. ЦИФРОВЫЕ ТЕХНОЛОГИИ

Цифровое предприятие (Digital Enterprise) — организация, которая использует информационные технологии (ИТ) в качестве конкурентного преимущества во всех сферах своей деятельности: производстве, бизнес-процессах, маркетинге и взаимодействии с клиентами.

Традиционная компания превращается в компанию с «цифровым мышлением», проходя путь цифровой трансформации (Digital Transformation).

В настоящее время выделяют девять основных элементов внедрения цифровых технологий в деятельность предприятий, которые в свою очередь можно объединить в три главных направления [1]. Рассмотрим их.

1. Использование цифровых информационных технологий в работе с клиентами:

а) Удовлетворенность клиентов продукцией (услугами) организации. Для получения информации используются социальные сети и цифровые СМИ.

б) Внедрение электронных программ гаджетов. Данный элемент характеризуется применением цифровых технологий для обеспечения дополнительных коммуникаций с клиентами (использование мобильных приложений для сбора информации и удаленного обслуживания)

в) Создание точек взаимодействия с покупателями продуктов и услуг обеспечивает суще-

ствующие и новые рабочие контакты с клиентами (аккаунты в соцсетях для оперативных ответов на жалобы и вопросы).

2. Трансформирование операционных процессов.

а) Цифровая автоматизация.

б) Виртуализация рабочей зоны. В данном случае подразумевается внедрение режима удаленной работы.

в) Оптимизация работы управленческого персонала. Внедрение информационных систем постановки задачи, сбора и обработки статистической информации менеджеры организации получают возможность опираться на действительные статистические данные.

3. Преобразование бизнес-моделей.

а) Применение цифровых преобразований в организациях (замена бумажных журналов (дублирование) электронными версиями в сети Интернет).

б) Открытие цифровых форм бизнеса (предоставление комплекса страховых услуг посредством работы интернетпортала).

в) Цифровая глобализация. Используя цифровые ресурсы (например, облачные технологии), компании могут оперативно принимать решения для любой географической точки своего бизнеса.

II. ЗАКЛЮЧЕНИЕ

Таким образом, процесс внедрения цифровых информационных технологий способствует появлению новых возможностей оптимизации бизнеса и получению конкурентных преимуществ.

СПИСОК ЛИТЕРАТУРЫ

1. Панфилова, Е. Е. Управление промышленной организацией в условиях цифровой экономики / Е. Е. Панфилова // Актуальные вопросы управления: Вестник университета, 2018. — № 11. — С. 23-28.

СЕТЕВЫЕ РЕШЕНИЯ РАЗВИТИЯ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРОЭНЕРГЕТИЧЕСКИМИ СИСТЕМАМИ

Алина Г. Ж., Марков А. В., Ташатов Н. Н.

Кафедра Компьютерной и программной инженерии, Факультет информационных технологий,
Евразийский национальный университет им.Л.Н.Гумилева

Нур-Султан, Республика Казахстан

Кафедра систем управления, Факультет информационных технологий и управления, Белорусский
государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: diamondgaxa.gzh@gmail.com

Ранее схема управления электроэнергетическими системами на основе цифровых технологий осуществлялась путем автоматизации существующих бизнес-процессов и технологических процессов. При всем этом на основе имеющихся цифровых технологий поддающиеся алгоритмизации процедуры постепенно переносились на цифровые устройства, которые были автоматизированы (с участием человека), а по возможности выполнялись автоматически (без участия человека) управляли оборудованием в соответствии с заданными алгоритмами и параметрами работы различных энергообъектов и систем в целом.

ВВЕДЕНИЕ

В данной статье предлагается рассмотреть технологии мониторинга и диагностики электрических сетей, понятие цифровых подстанций, систему управления интеллектуальными энергетическими системами.

I. ЦИФРОВЫЕ ПОДСТАНЦИИ

Рассмотрим и сравним технологии мониторинга и диагностики электрических сетей [1]. Под «цифровой» подстанцией (ЦПС) понимается подстанция с высоким уровнем автоматизации управления, в которой практически все процессы информационного обмена между элементами ПС, обмена с внешними системами, а также управления работой ПС осуществляются в цифровом виде на основе протоколов МЭК. При этом и первичное оборудование цифровых подстанций, и компоненты информационно-технологических и управляющих систем функционально и конструктивно направлены на поддержку цифрового обмена данными. Прямыми целями создания ЦПС являются:

- улучшение мониторинга и управления электросетевым оборудованием ПС;
- увеличение надежности работы и эффективности эксплуатации оборудования ПС путем формирования и унификации, основных информационно-технологических и управляющих систем (ИТС),
- переход к «необслуживаемым» подстанциям, т.е. к подстанциям без непрерывного дежурства оперативного персонала. Переход к трансляции сигналов в цифровом виде на всех уровнях управления ПС позволит получить целый ряд преимуществ, в том числе:

- существенно сжать затраты на физические вторичные цепи и каналы их прокладки;
- увеличить помехоустойчивость вторичных цепей благодаря переходу на цифровую связь с применением для передачи медных кабелей, а при больших расстояниях, больших скоростях и неблагоприятной электромагнитной обстановке – оптоволоконной среды;
- упростить и удешевить конструкцию микропроцессорных устройств за счет исключения трактов ввода аналоговых сигналов;
- унифицировать интерфейсы устройств, существенно упростить взаимозаменяемость этих устройств (в том числе разных производителей);
- унифицировать процессы проектирования, внедрения и эксплуатации подстанции. [2]

Для вертикально организованной системы от крупных генерирующих объектов до потребителей такой подход полностью обеспечил эффективное и надежное управление режимами, а также работу механизмов рынка электроэнергии.[3]

II. ТЕХНОЛОГИИ МОНИТОРИНГА И ДИАГНОСТИКИ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ

Рассмотрим используемые в данное время системы мониторинга. Системы и устройства диагностики и мониторинга состояния предназначены для получения, сбора и обработки информации о техническом состоянии оборудования, что позволит прогнозировать не только объемы и сроки ремонтов, но и сроки службы оборудования, что в конечном счете вносит вклад в повышение надежности и безаварийности энергоснабжения потребителей. Технологии мониторинга и диагностики должны осуществлять автоматизированный сбор данных, передачу их по каналам связи (беспроводным, оптическим), об-

работку, анализ и выдачу информации на диспетчерские пункты о состоянии контролируемых параметров. В настоящее время для мониторинга и диагностики используется:

- воздушное лазерное сканирование;
- наземное лазерное сканирование;
- мониторинг температуры нагрева проводов с помощью установленных на проводах датчиков температуры;
- мониторинг токовой нагрузки, скорости ветра, температуры, габарита в точке установки;
- мониторинг гололедной обстановки с помощью;
- мониторинг грозовой активности;
- мониторинг силовых трансформаторов. [4]

Экологические и экономические вызовы современного мира стали причиной создания и внедрения новых технологических решений прежде всего в распределительной части электроэнергетических систем — возобновляемых источников электроэнергии (ВИЭ), имеющих слабо регулируемый характер генерации и широкий диапазон мощностей, накопителей электроэнергии (НЭЭ) различного назначения, устройств силовой электроники для активного управления потоками мощности в сетях переменного тока и построения сегментов распределительных сетей, работающих на постоянном токе. Эти и другие устройства все шире и шире распространяются в по всему полю распределительных электросетей. В результате электрическая сеть стала все больше превращаться в мультиагентную систему, потоки энергии в которой могут быть направлены в любых направлениях — вниз, как и раньше, а также вверх и/или по горизонтали, т. е. между равноправными участниками любого уровня этой сети. Такие изменения в технологической части энергосистемы неизбежно влекут за собой изменения бизнес-процессов и механизмов рынка электроэнергии. Как следствие, сложность системы многократно возрастает и в ней классические решения по автоматизации оказываются как минимум не эффективными. Развиваемые системы технологического управления строятся как распределенные иерархические системы. Средства автоматизации, связи и вычислительной техники должны координировано функционировать на разных уровнях иерархии: на энергообъектах и в центрах управления. Роль устройств нижнего уровня иерархии управления играют объектные программно-технические средства систем (подсистем) сбора, обработки и передачи оперативной информации и неоперативной телеметрической информации. Кроме того, средствами автоматизированных систем управления электроэнергетических подстанций, обеспечивается возможность непосредственного управления оборудованием объектов из удаленных центров управления основных функций. Указанные системы осуществляют поддерж-

ку и обеспечивают автоматизацию основных функций оперативно-диспетчерского управления оперативно-технологического управления, а также процессов управления функционированием и эксплуатацией сетей. Использование средств современных информационных технологий позволяет реализовать на каждом уровне управления систем автоматического и автоматизированного управления технологическими процессами совместное использование оперативной и ретроспективной информации, накапливаемой и используемой в каждой из систем. Это дает возможность учета ограничений, более адекватного прогноза состояния объекта управления, и, соответственно, повышения качества управления. При этом общими являются ограничения по надежности электроснабжения потребителей и её составляющих.[5] Интеллект в энергетических станциях в значительной степени определяется системой управления. Основой управления режимом электроэнергетических станций в целом является оперативно-диспетчерское управление, выполняемое персоналом, и оперативно-технологическое управление, выполняемое персоналом. Автоматизированная система диспетчерско-технологического управления, в среде которой действует оперативный персонал, поддерживает информационную модель объекта управления с системой удобного отображения состояния объекта, системой передачи и реализации управляющих воздействий (включая телеуправление); система включает программное обеспечение (ПО), автоматизирующее процессы принятия решения, ПО для анализа, планирования режимов, подготовки решений по настройке автоматических контуров управления и др.

ЗАКЛЮЧЕНИЕ

Автоматизация перечисленных задач и внедрение соответствующих сетевых решений существенно облегчит анализ текущей ситуации, предоставляя диспетчерскому персоналу информацию об имеющихся запасах надежности и ресурсах по управлению - для принятия, при необходимости, мер по устранению нарушений или их угрозы.

1. Наумов В.А. От автоматизированной энергетики к интернету энергии / В.А. Матисон. [Электронный ресурс]/ Режим доступа : <https://energybase.ru/> Дата доступа : 14.10.2020
2. Мастерова О.А. Эксплуатация электроэнергетических систем и сетей / Барская А.В.: Учеб. пособие. Томск: Издательство ТПУ, 2006
3. «Presentation of recent and ongoing R&I projects in the scope of the ETIP SNET Part1. Final» [Электронный ресурс]: ETIP SNET, 2018, 103 с.
4. «Архитектура Интернета энергии. Предварительная версия». [Электронный ресурс] Инфраструктурный Центр EnergyNet, 2018, 57 с.
5. Taft J., «Ultra Large-Scale Power System Control Architecture. A Strategic Framework for Integrating Advanced Grid Functionality»./ P. De Martini CISCO, 2012, 25 с.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗНОВИДНОСТЕЙ РОБОТОВ ДЛЯ АВТОМАТИЗАЦИИ ПИЩЕВОГО ПРОИЗВОДСТВА

Архипенко Я. С., Парамонова А. Е., Белощедов Е. С., Гаруля Д. В.

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: arhipenko.yana.serg@mail.ru

Вопросу автоматизации производства пищевой промышленности в настоящее время уделяется огромное внимание, т.к. отрасль производства продуктов играет ведущую роль в жизни общества. Для обеспечения комфортной жизни необходимо обеспечить людей достаточным количеством продуктов питания, предусмотреть разнообразие товаров, доступные цены и соблюсти нормы качества.

Формирование стоимости единицы продукции достаточно сложный процесс, который включает в себя затраты на сырье, работу персонала, транспортные расходы, рекламу и другое. Для покупателей, как правило, не принципиально из чего именно складывается окончательная цена товара, важно, чтобы итоговая сумма была удовлетворительной. Изготовителям важно сокращать затраты на каждом этапе производства, оптимизировать работу, минимизировать время выполнения операций, уменьшать количество бракованной продукции, за счет оптимизации данных параметров можно добиться снижения стоимости товара.

Существует несколько методов оптимизации производства. Основными являются «Бережливое производство», при котором сокращается количество запасов готовой продукции, уменьшаются объёмы материалов. Второй метод – «Тотальная оптимизация» – метод, при котором к процедуре по снижению затрат привлекаются непосредственно рабочие, учитывается их мнение, опыт, знания, идеи. [1]

Рациональное использования труда операторов, отказ от выполнения рутинной, однотипной работы может существенно повысить показатели производства. Добиться замены ручного труда можно путем внедрения на производственные линии роботов, которые смогут помочь человеку.

Определённый тип работ предусматривает выполнение конкретного ряда задач, которые предусмотрены на различных линиях.

Процесс производства практически любого типа продовольственных товаров включает в себя много стадий, которые можно детально описывать достаточно долго и подробно. Нас интересуют стадии, на которых работу оператора можно заменить на работу робота. Т.е. подразумевается работа на линиях-конвейерах. Рутинная работа операторов может быть заменена роботами в случаях точечного удаления продукции с конвейера – проверка на качество упаковки, печать маркировки и т.д. (шаг 1), укладки готовой

продукции в коробки (шаг 2), перемещении коробок на подвижную конструкцию (шаг 3), непосредственная транспортировка груза (шаг 4).

Действию, которое требует особую точность выполнения, лучше доверить роботу с не самой высокой скоростью, но улучшенным позиционированием и точной траекторией движения, например, роботам-манипуляторам типа SCARA (Selective Compliance Articulated Robot Arm). Это манипуляторы с селективной гибкостью. Конструктивно они жесткие в вертикальной плоскости, то есть вдоль оси Z, при этом в горизонтальной плоскости (по осям X и Y) обладают податливостью. Такие роботы часто выполняют сборочные операции. Роботы типа SCARA могут работать быстрее, чем декартовые роботы, и имеют небольшие габариты, но они могут быть более дорогостоящими. Применение SCARA роботов особенно выгодно для сборки узлов, где робот должен вкладывать одни детали в другие, при этом, не соединяя их. Важным является то, что благодаря своей конструкции, манипулятор может вытянуться, распрямив «локоть», а может свернуться, освободив занимаемое пространство. Это удобно при работе в ограниченном пространстве, и когда детали перемещаются из одного производственного модуля в другой. Часто возникает необходимость очень быстро и аккуратно складывать штучную готовую продукцию в тару для упаковки. В таких случаях поможет Дельта-роботы – это один из видов параллельных роботов, отличительной особенностью которых является треугольная платформа с тремя шарнирными рычагами. Особенностью является использование параллелограммов в конструкции манипулятора, что позволяет сохранять пространственную ориентацию исполнительного устройства робота. Основным преимуществом дельта-роботов является их высочайшая скорость перемещения за счет минимальной инерции.

Линейные роботы позволяют подготовить груз для транспортировки, удобно сложить коробки с готовой продукцией, переместить к необходимой отправной точке для дальнейшей транс-

портировки. Декартовы роботы имеют, как правило, три линейные оси управления. Каждая из этих осей находится под прямым углом к двум другим. Если одно из звеньев, которое совершает горизонтальное перемещение, имеет поддержку на обоих концах звена, то такой декартов робот называется порталным. Так как декартовы роботы имеют только линейные перемещения, то разработчикам достаточно просто написать программу для перемещения манипуляторов в любую точку пространства, используя несложные тригонометрические функции. Характеристики порталных роботов могут быть совершенно разнообразными и зависят от выбранных линейных сервоприводов и механической части.

Транспортировку грузов удобно выполнять используя промышленные мобильные роботы с большой грузоподъемностью. Внедрение в них датчиков и средств навигации в сочетании с развитым алгоритмическим обеспечением обеспечивают их высокой скоростью и гибкостью применения. Они могут быть интегрированы в другие системы, обладающие возможностью движения, и иметь свою автономную систему навигации. Например, автономные роботы способны тянуть за собой группу тележек и удерживать грузы. Они работают с разными типами пневматических тягово-сцепных соединителей. Оборудованы функциями безопасности, которые позволяют мобильному роботу автономно и безопасно перемещаться по территории производственных помещений.

Наиболее интересным представителем «семейства» роботов является коллаборативный робот. Он способен работать непосредственно рядом с человеком за счет особых механизмов, позволяющих блокировать и предотвращать нежелательные манипуляции. Робот может распознавать и контролировать приближение человека.

При нарушении безопасной зоны, робот автоматически прекращает работу, что позволяет избежать каких-либо непредвиденных травм для оператора. Коллаборативные роботы оснащены датчиками, ограничивающими усилие и/или скорость звеньев, и, в зависимости от применения, могут работать в непосредственной близости от человека без установки защитного ограждения. Некоторые из таких роботов могут быть «двурукими», чтобы лучше копировать манипуляционные способности человека и легче интегрироваться в существующий производственный процесс без необходимости его перестраивать. Адаптивная точность коллаборативных роботов позволяет им эффективно работать в полуструктурированных средах, используя встроенную систему машинного зрения. [2]

Проанализировав назначение каждого робота, изучив план, последовательность выполнения операций на линии по производству пищевой продукции, можно сделать вывод, что для оптимизации производства на первом шаге работу оператора можно заменить на более эффективную работу коллаборативного робота, на шаге 2 подойдет Дельта-робот, для шага 3 оптимально применение линейного робота, на шаге 4 (транспортировка готовой продукции) используют мобильные роботы большой грузоподъемности.

Каждый тип робота представлен на рис. 1.

СПИСОК ЛИТЕРАТУРЫ

1. Джеймс Вумек, Бережливое производство: Как избавиться от потерь и добиться процветания вашей компании / Джеймс Вумек, Дэниел Джонс – 12-е издание, – Москва : Альпина паблишер, 2018. – 650 с.
2. СИСТЕМЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ [Электронный ресурс] / РОБОТИЗАЦИЯ И АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВА. – Минск, 2020. – Режим доступа : <http://sia.by/typy-promyshlennyh-robotov-i-ih-funkczii/>. – Дата доступа : 15.10.2020



Рис. 1 – Разновидности роботов

ИСПОЛЬЗОВАНИЕ БЕССЕРВЕРНЫХ ТЕХНОЛОГИЙ ДЛЯ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Волосович С. В.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: svalasovich@gmail.com

ВВЕДЕНИЕ

Данные играют важную роль в машинном обучении, анализе данных, науке о данных, их представление и качество являются первостепенной заботой, поэтому обработка и управление данными играют важную роль.

Обработка данных, чаще всего, представляет собой высоконагруженный процесс, который требует высокие мощности, дорогостоящее оборудование, установку и настройку инструментов для реализации распределенной обработки данных (например Apache Spark, Apache Flink, Apache Hadoop, Apache Samza, Nemo), настройку и обслуживание серверов. При пакетной обработке данных в большинстве случаев работа выполняется по расписанию, либо другой сервис является триггером для запуска процесса обработки, это означает, что часть времени кластер простаивает без работы.

Для таких приложений можно использовать подход, при котором не требуется приобретать и настраивать оборудование, а использовать облачные бессерверные сервисы, где оплата производится только за использование ресурсов.

I. БЕССЕРВЕРНЫЕ ВЫЧИСЛЕНИЯ, ПРИЛОЖЕНИЯ И ОСОБЕННОСТИ БЕССЕРВЕРНЫХ СЕРВИСОВ

Бессерверные вычисления — естественная для облака архитектура, которая позволяет передать большую часть операционной ответственности на провайдера. Бессерверные вычисления позволяют создавать и запускать приложения и сервисы, не беспокоясь о серверах. Они устраняют необходимость заниматься вопросами управления инфраструктурой — такими, например, как выделение серверов или кластеров, необходимых ресурсов, а также установка исправлений и обслуживание операционной системы.

Бессерверное приложение — приложение которое не требует выделения, обслуживания и администрирования серверов для таких компонентов, как вычислительные ресурсы, базы данных, хранилища, компоненты обработки потоков и организации очередей сообщений и т. д. Заботиться об обеспечении отказоустойчивости и доступности приложения не требуется, этим занимается поставщик услуг.

Провайдеры облачных услуг предоставляют набор полностью управляемых бессерверных сервисов (сервисы для вычислений, хранилища,

хранилища данных, оркестраторы, сервисы для аналитики и т.д.), которые можно использовать для создания и работы бессерверных приложений. Особенности бессерверных сервисов:

- Абстракция. Не требуется управление сервером на котором запускается сервис. Клиент ничего не знает про операционную систему, сетевые настройки и прочее;
- эластичность. Поставщик бессерверных сервисов предоставляет необходимое количество вычислительных ресурсов, в зависимости от того, какая нагрузка приходится на ваше приложение;
- эффективная стоимость. Если ваше приложение не выполняет работу, то вы не платите, т.к. оно не потребляет вычислительные ресурсы;
- ограниченный жизненный цикл. Если ваше приложение не используется, то сервис автоматически его останавливает.

II. БЕССЕРВЕРНЫЕ СЕРВИСЫ ДЛЯ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Приведем список бессерверных сервисов для обработки больших данных и их поставщиков:

- Dataflow — Google Cloud;
- AWS Glue – Amazon Web Services;
- Azure Data Factory — Microsoft Azure.

Для примера подробнее рассмотрим Dataflow. Google Cloud Dataflow – это управляемая бессерверная служба для выполнения пакетной или потоковой обработки данных. Для написания Dataflow задач используется Apache Beam SDK.

После написания pipeline с помощью Apache Beam, можно использовать Dataflow для его выполнения. Попав в сервис, pipeline становится Dataflow задачей. Dataflow сервис полностью управляет сервисами Google Cloud, такими как Compute Engine и Cloud Storage для выполнения задания Dataflow, автоматически выполняет все функции администрирования, включая создание кластера, масштабирования необходимых ресурсов, мониторинг результата и ведение журнала. Для выполнения задания, Dataflow разбивает задание на трансформации и строит граф выполнения (см. рис. 1), после чего согласно построенному графу выполняются все трансформации. По окончании задания, Dataflow автоматически останавливает кластер и освобождает все ресурсы.

Построение графа занимает некоторое время, поэтому для часто используемых заданий рекомендуется создавать Dataflow Template, при создании которого трансформации с построенным графом записываются в файл и может быть сохранено в Google Storage. После этого Dataflow задание может быть запущено без необходимости повторного построения графа.

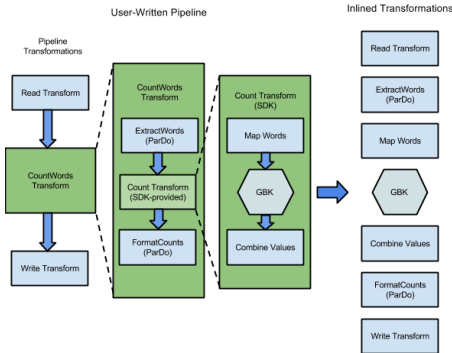


Рис. 1 – Построение графа выполнения на примере подсчета слов

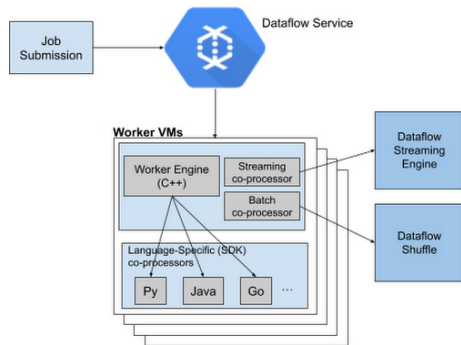


Рис. 2 – Архитектура worker'а

Вычислительным узлом в Dataflow является worker (см. рис. 2), который имеет виртуальные CPU (vCPU) и оперативную память в зави-

симости от заданного типа Compute Engine. Максимальное и стартовое количество worker'ов для выполнения задачи можно задавать при запуске.

III. СРАВНЕНИЕ ВОЗМОЖНОСТЕЙ

Сравним бессерверные сервисы для обработки данных от разных поставщиков и представим в виде таблицы (см. таблицу 1). Для сравнения будем рассматривать следующие параметры:

1. Название сервиса;
2. возможность пакетной обработки;
3. возможность потоковой обработки;
4. учитываемые параметры для расчета стоимости выполнения задания;
5. наличие и тип SDK;
6. языки программирования для разработки задания по обработке данных;
7. визуализация выполняемых этапов задачи;
8. программное ядро.

Как видно из таблицы, базовые потребности в обработке данных удовлетворяет каждый сервис. Однако из-за разного программного ядра и разных политик ценообразования, в целях экономии и уменьшения времени исполнения, к каждому заданию требуется индивидуальный подход, для выбора необходимого сервиса.

СПИСОК ЛИТЕРАТУРЫ

1. AWS Glue Documentation [Electronic resource] / Amazon. – Amazon Web Services. – Mode of access: <https://docs.aws.amazon.com/glue/>. – Date of access: 18.10.2020.
2. Dataflow Documentation [Electronic resource] / Google. – Google Cloud. – Mode of access: <https://cloud.google.com/dataflow/docs/>. – Date of access: 18.10.2020.
3. Azure Data Factory Documentation [Electronic resource] / Microsoft. – Microsoft Azure. – Mode of access: <https://docs.microsoft.com/en-us/azure/data-factory/>. – Date of access: 18.10.2020.

Таблица 1 – Сравнительная характеристика

№ Параметра	Amazon Web Services	Google Cloud	Microsoft Azure
1	AWS Glue	Dataflow	Azure Data Factory
2	Да	Да	Да
3	Да	Да	Да
4	Время затраченное на выполнение задания, количество DPU (вычислительный узел)	Время затраченное на выполнение задания, количество worker'ов, тип worker'а, объем перемешивания (shuffle)	Оплата за вызов задания, время выполнения задания, используемое ядро для трансформаций и тип вычислительного узла
5	Apache Beam	Собственная реализация на базе Apache Spark	Собственная реализация
6	Java, Python, Go	Scala, Python	Code-free (не требует знания языков программирования), .NET, Python
7	Да	Да	Да
8	Собственная реализация на C++	Apache Spark	Зависит от выбранной среды исполнения трансформаций

НАСТРОЙКА ПИД-РЕГУЛЯТОРА ЧЕРЕЗ АПЕРИОДИЧЕСКИЙ ЦИФРОВОЙ РЕГУЛЯТОР

Гринюк Д. А., Олиферович Н. М., Сухорукова И. Г., Оробей И. О.

Кафедра автоматизации производственных процессов и электротехники, Белорусский государственный технологический университет
Минск, Республика Беларусь
E-mail: hryniukda@gmail.com

Рассматривается метод настройки ПИД-регулятора. Для этого предлагается использовать deadbeat регулятор с различными вариантами синтеза. В основе синтеза лежит аппроксимация переходной характеристики deadbeat регулятора ПИД-регулятором. Предложенные варианты позволяют легко настроить замкнутую систему регулирования на различные качества переходного процесса. В качестве параметра вариации выступает время квантования цифрового регулятора. Используя только один данный параметр можно подстраивать все три параметра настройки ПИД-регулятора. Получаемые в результате настройки оказываются близкими к интегральным критериям.

ВВЕДЕНИЕ

ПИД-регулятор продолжает оставаться одним из популярных решений для построения систем управления. Его универсальность для широкого класса объектов способствует широкому использованию как в технологических процессах в промышленности, так и в технических электронных системах. За время векового применения предложено множество решений по развитию классического метода построения ПИД-регулирования. Одновременно происходит и развитие методов настройки регуляторов данного типа. Основными методами настройки являются коэффициент усиления K , время интегрирования I и дифференцирования D передаточной функции регулятора.

I. МЕТОДЫ НАСТРОЙКИ

В литературе предложено большое количество методов настройки. Среди них можно выделить два основных полюса. Один полюс – это обеспечение требуемого запаса по устойчивости, другой – обеспечение желаемого качества переходного процесса. Большинство наиболее популярных методик настройки ПИД-регулятора используют один из полюсов как отправную точку, а затем, по необходимости, обеспечивают компромисс в отношении другого полюса. Среди популярных критериев устойчивости можно выделить частотные (критерий Найквиста и Михайлова) и критерии, основанные на анализе характеристического уравнения передаточной функции канала управления (корневой критерий, критерий Стодолы и Гурвица). Соответственно, среди частотных методов можно выделить модифицированный метод Ziegler-Nichols, Takahashi, метод расширенных частотных характеристик, использование логарифмических частотных характеристик. Принцип настройки основан на обеспечении декремента затухания переходного процесса. На основе данного подхода

базируется также метод Schaedel. Корневой метод настройки лежит в основе метода Дудникова. На обеспечении аperiodического переходного процесса основан метод настройки амплитудного оптимума и метод Skogestad. Сюда можно отнести метод Cohen-Coon с нахождением эквивалентной передаточной функции и метод с использованием таблиц Chien-Hrones-Reswick. Однако выше указанные методы для обеспечения качества переходного процесса требуют последующей коррекции настроек. По этой причине, а также с широкими возможностями современных компьютеров и контроллеров происходит развитие онлайн методов анализа настройки с использованием различных поисковых методов. Среди них можно выделить как методы, основанные на классических подходах минимизации интегральных критериев, так и широкое применение нечетких и генетических алгоритмов. Эти методы более универсальны, могут применяться для линейных и нелинейных объектов управления. С помощью онлайн методов настраиваются нелинейные регуляторы. Однако они требуют высокой вычислительно мощности.

II. ОФОРМЛЕНИЕ ТЕКСТА

В [1] предложено осуществить настройку ПИД-регулятора через deadbeat (в русскоязычной литературе встречаются различные варианты перевода: аperiodический, компенсационный и т.д.). Однако были отмечены ограничения этого подхода. Этот метод прямого проектирования для дискретных ПИД-регуляторов может представлять интерес для следующих случаев: 1. Применение самонастраивающегося управления для уникальной настройки параметров ПИД-контроллеров. 2. Определение подходящих начальных значений для оптимизации числовых параметров. Предложенный в [1] подход основан на анализе коэффициентов передаточной функции объекта управления. Однако это не применимо, если мы используем для синтеза deadbeat

регулятор с ограничением на управляющее воздействие [2]. Для нахождения коэффициентов K , I , D можно воспользоваться прямой аппроксимацией передаточной функции deadbeat регулятором ПИД регулятора. Построение импульсной характеристики цифрового регулятора по его передаточной функции

$$W_{DB}(z) = \frac{p_1 z^{-1} + \dots + p_m z^{-m}}{q_0 + q_1 z^{-1} + \dots + q_m z^{-m}} \quad (1)$$

не требует решения дифференциального уравнения. В (1) z – переменная z -преобразования $z = \exp(T_0 s)$; T_0 – время квантования; m – порядок полинома. Значения на каждом такте может быть найдено путем простейших арифметических операций. По переходной характеристики deadbeat, когда градиент приращения становится постоянным, можно по двум соседних значения управления u и вычислить время интегрирования:

$$I = \frac{u(k+1) - u(k)}{T_0}, \quad (2)$$

где k – дискретные отсчеты $k = t/T_0 = 0, 1, 2, \dots$; t – время. Следующим этапом можно вычислить коэффициент усиления регулятора K

$$K = u(k) - I k T_0. \quad (3)$$

Время дифференцирования зависит от формы записи регулятора. Для идеального регулятора формула следующая

$$D = q_0 - K. \quad (4)$$

В случае формы реального регулятора, как наиболее интересной с практической точки зрения

$$W_R(s) = K + \frac{1}{I s} + \frac{D s}{F s + 1}. \quad (5)$$

Тогда расчетная формула будет

$$D = \frac{q_0 - K}{F}. \quad (6)$$

Для вариации настроек можно воспользоваться изменением времени T_0 и величины первичного управляющего воздействия на различном количестве тактов N [2]. В качестве критерия выбора можно использовать различные интегральные критерии. При этом полученные настройки ПИД-регулятора не требуют обязательного использования simple time, который использовался для синтеза deadbeat (DBC).

III. СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЗУЛЬТАТОВ НАСТРОЙКИ

Предложенная методика настройки была опробована для некоторых передаточных функций. За основу была взята передаточная функция из [1]. Три другие имели отличную динамику от базовой: другое время запаздывания или вид в

числителе. Качество настройки будем оценивать с помощью интегральных критериев[3]:

$$\int_0^{tf} e(t)^2 dt \rightarrow \min; \quad (7)$$

$$\int_0^{tf} t^2 |e(t)| dt \rightarrow \min; \quad (8)$$

$$\int_0^{tf} u(t)^2 dt \rightarrow \min; \quad (9)$$

$$\int_0^{tf} (e(t)^2 + u(t)^2) dt \rightarrow \min; \quad (10)$$

где $e(t)$ – отклонение выхода сигнала задания; $u(t)$ – сигнал управления на выходе регулятора; tf – время моделирования. Для моделирования использован непрерывный ПИД-регулятор с $F = 0,01$. Время переходного процесса оценивалось по уровню 3 процента от сигнала задания. Также, для сравнения проведена численная оптимизация интегральных критериев в программном пакете MatLAB.

Более подробно с результатами исследования можно ознакомиться [4,5]

IV. ЗАКЛЮЧЕНИЕ

Предложенная методика настройки ПИД-регуляторов через DBC оказалась очень удобной с практической точки зрения. Вариацией параметра T_0 можно сбалансировано менять одновременно три настройки ПИД-регулятора. Все выбранные в исследовании параметры переходного процесса имеют один глобальный минимум в зависимости от T_0 . Для интегральных критериев, которые ориентированы только на выходной параметр объекта управления (9) и (10), использование предложенной методики может служить отправной точкой для дальнейшего использования в численных методах оптимизации. При проведении представленных исследований оптимизация начальных настроек через DBC позволила получить результат за меньшее количество итераций, чем ранее с использованием метода Ziegler-Nichols.

1. Isermann R. Digital Control Systems. Springer-Verlag. 1989, 565 p.
2. Increasing the robustness of the digital controller / [N. Olfierovich. et al.] // 2018 Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, 2018, pp. 1–6.
3. Non-linear PID controller and methods of its setting. [D. Hryniuk et al.] // 2017 Open Conference of Electrical, Electronic and Information Sciences (eStream), 2017, pp.1–4.
4. Approximation PID-Controllers Through Deadbeat Controller and its Tuning [D. Hryniuk et al.] // 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), 2019, pp.1–6.
5. Метод настройки ПИД-регулятора через deadbeat регулятор на различные интегральные критерии [Гринюк Д. А. и др.] // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. 2019. в печати.

МЕТОДЫ СИНТЕЗА ШИРОКОПОЛОСНЫХ СОГЛАСУЮЩИХ УСТРОЙСТВ К ИЗМЕНЯЮЩЕМУСЯ ИМПЕДАНСУ НАГРУЗКИ

Дубовик И. А., Бойкачев П. В., Исаев В. О.

Учреждение образования «Военная академия Республики Беларусь»

Минск, Республика Беларусь

E-mail: {duba-77, pashapash.boi}@mail.ru, ystasmoz@gmail.com

В настоящее время широкое применение нашли методы синтеза широкополосных согласующих цепей, где в качестве нагрузки рассматриваются антенные, усилительные и другие устройства, представленные в виде значения комплексного сопротивления на дискретном ряде частот. Но в то же время в методах синтеза ШСЦ не учитывается непостоянство комплексного сопротивления нагрузки РТУ, вызванное изменением окружающей среды либо условием эксплуатации. Таким образом в данной работе был выполнен обзор с последующим анализом существующих методов синтеза к изменяющемуся импедансу нагрузки.

ВВЕДЕНИЕ

В настоящее время широкое применение нашли методы синтеза широкополосных согласующих устройств (ШСУ), где в качестве нагрузки радиотехнических устройств (РТУ) рассматриваются антенные, усилительные и др. устройства, обладающие некоторым комплексным сопротивлением. Но при этом в методах синтеза ШСУ, не учитывается непостоянство комплексного сопротивления нагрузки, вызванное изменением условий их эксплуатации. Изменение величины комплексного сопротивления нагрузки приводит к рассогласованию тракта РТУ с нагрузкой. Это способствует появлению в тракте отраженной волны, что приводит к потерям мощности передаваемого (принимаемого) сигнала. Таким образом, необходимо оценить возможности применения существующих методов синтеза к синтезу ШСУ в задачах адаптации РТС к возмущающим воздействиям на них.

I. ОСНОВНАЯ ЧАСТЬ

Исходя из поставленной задачи, был выполнен обзор и анализ (по ряду критериев [1]) методов синтеза ШСУ. По итогам проведенного обзора было установлено:

1. Использование аналитических методов синтеза [2-4] позволяет найти решение задачи синтеза ШСУ для нагрузок, имеющих невысокий порядок и представленных в виде эквивалента ограничиваясь при этом выбранной аппроксимирующей функцией В то же время, современные РТУ представляются в виде сложных нагрузок [5], поиск эквивалента которых является непростой задачей. В связи с этим, использование аналитической теории для синтеза ШСУ, с учетом изменяющегося импеданса нагрузки, нецелесообразно, что подтверждается результатами обзора.

2. Параметрические и структурно-параметрические методы синтеза [6-7] всегда приводит к определенному результату, но они

зависят от выбора начального приближения и способа формирования целевой функции. Существует проблемы сходимости, возможность получения только локальных оптимумов.

3. Использование графоаналитического метода синтеза [8] приводит к получению простых Г – или L – образных согласующих цепей, нагруженной на комплексное сопротивление нагрузки. Методика является трудоемкой при расчете ШСУ. Для анализа методов синтеза были синтезированы ШСУ, обеспечивающие максимальный уровень передачи мощности в нормированном диапазоне частоты (от 0 до 1 Гц). В качестве сопротивления генератора использовалось активное сопротивление Ом, а в качестве сопротивления нагрузки использовались нагрузки различных классов [3], отличающиеся друг от друга расположением нуля передачи. Результаты синтеза ШСУ представлены в таблице 1.

II. ЗАКЛЮЧЕНИЕ

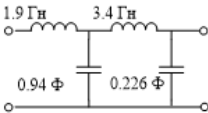
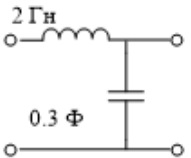
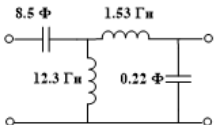
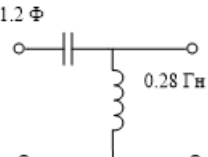
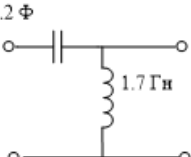
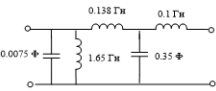
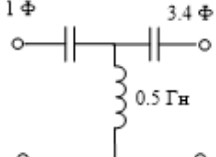
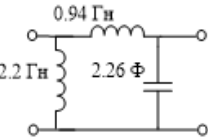
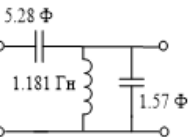
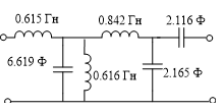
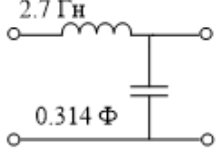
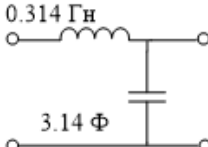
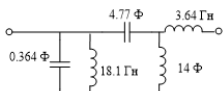
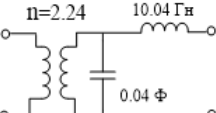
По результатам проведенного анализа было установлено, что использование структурно-параметрических методов синтеза позволяют синтезировать ШСУ, которые обеспечивают оптимальное значение целевой функции. Но так, как структура цепи известна лишь на последнем этапе синтеза, то данные ШСУ не будут обеспечивать минимальную чувствительность. Исходя из этого, для решения задачи синтеза ШСУ с учетом влияния изменения параметров нагрузки, предлагается использовать параметрический метод синтеза. Это обусловлено тем, что в параметрических методах синтеза структура ШСУ известна, а использование комбинационного подхода, обеспечивать как высокий уровень КПМ, так и минимальную чувствительность.

СПИСОК ЛИТЕРАТУРЫ

1. Ланнэ, А.А. Оптимальный синтез линейных электрических цепей / А.А. Ланнэ. - М.: Связь, 1969.-423с.
2. Youla, D.C., A new theory of broadband matching. IEEE Trans. 1964; 11(1): 30-50.

3. Chen, W.K., Synthesis of optimum Butterworth and Chebyshev broad-band impedance-matching networks. IEEE Trans. 1977; 5(24): 157-169.
4. Филиппович, Г.А., Широкополосное согласование сопротивлений. Минск;2004.
5. Yarman, B.S., Design of Ultra Wideband Power Transfer Networks. Hoboken, NJ: Wiley, 2010.
6. Carlin H., Amstutz P., Broadband Matching. Anales des telecommunications. IEEE Trans. 1981; 28 (5):401-405.
7. Yarman, B.S. Real frequency broadband matching using linear programming. RCA Rev. 1982; 43 (4):626-654.
8. Курушин А.А, ред: Б.Л. Когана. Проектирование СВЧ устройств с использованием электронной диаграммы Смита. М: МЭИ; 2008.

Таблица 1 – Результаты синтеза ШСЦ

Класс нагрузки	Метод вещественных частот	Графоаналитический метод Вольперта-Смитта	Структурно-параметрический метод на основе аппарата Т-матрицы	Обобщенный метод Дарлингтона
I класс ($R_1=0.2\text{Ом}$, $R_2=3\text{Ом}$, $C=0.05\Phi$)				-
II класс ($R=1\text{Ом}$, $C=1.414\Phi$)				
III класс ($L_1=4\text{Гн}$, $L_2=0.75\text{ Гн}$, $L_3=0.3\text{Гн}$, $C=2\text{Гн}$)				
IV класс ($C=0.2\Phi$, $R=5\text{Ом}$)				-

АНАЛИЗ КАЧЕСТВА ПОКРЫТИЯ ТЕХНОЛОГИЧЕСКОГО УЧАСТКА ВАКУУМНОЙ УСТАНОВКИ

Железнёв А. И., Хаджинов М. К.

Кафедра систем управления, Белорусский государственный университет информатики и
радиоэлектроники

Минск, Республика Беларусь

E-mail: zhelezonsany@gmail.com

Рассматривается структура технологического участка вакуумной установки. Анализируется вопрос качества получаемого покрытия и методах его повышения. Выявлены экономически выгодные методы повышения качества покрытия подходящие для условий реального производства.

ВВЕДЕНИЕ

Метод вакуумного осаждения материала является одним из способов получения тонкопленочных покрытий. Для того, чтобы получить такие покрытия для массового использования в установках подобного рода применяется система непрерывной подачи тележек в зону напыления.

В такой системе обязательно присутствует участок, где происходит непрерывное осаждение материала на перемещающиеся тележки, которому уделяется особое внимание, поскольку именно благодаря процессам, происходящим в нем, на выходе получаем покрытие, готовое к дальнейшему использованию. Этот участок носит название технологическим, а соответствующая камера – технологической. В связи с этим для такого технологического участка вакуумной установки необходимо определить параметры, влияющие на качество получаемого покрытия.

I. ОПИСАНИЕ ТЕХНОЛОГИЧЕСКОГО УЧАСТКА

Технологический участок состоит из транспортной и осаждающей систем (см. рис. 1).

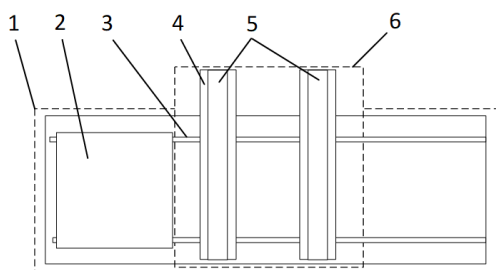


Рис. 1 – Структурная схема технологического участка (1 – транспортная система, 2 – тележка, 3 – направляющие, 4 – технологическая защита, 5 – устройства осаждения, 6 – система осаждения)

Следует отметить, что транспортная система состоит из тележки линейно перемещающаяся по направляющим. Система осаждения состоит из устройств осаждения материала, которые оснащены технологической защитой для предотвращения влияния устройства одной позиции на покрытие в другой. Задачей этого участка является осаждение материала на подложки, рас-

положенные на тележке и движущиеся с технологической скоростью по направляющим под устройствами осаждения.

II. ПАРАМЕТРЫ КАЧЕСТВА ПОКРЫТИЯ

Качество осаждения является одной из главных задач любого типа вакуумных установок. Под качеством осаждения понимают значения показателей, которые в наиболее полном объеме характеризуют качество покрытия. Такие параметры включают в себя:

- пористость;
- неравномерность;
- адгезионная способность.

Под пористостью понимают степень заполнения нанесенного покрытия порами. Показателем равномерности оценивают неизменность параметров нанесенного покрытия на подложки. Адгезионная способность же характеризует силу сцепления материала осаждения и напыляемые подложки [1].

III. МЕТОДЫ ПОВЫШЕНИЯ КАЧЕСТВА ПОКРЫТИЯ

В условиях производства, такие параметры качества покрытия как пористость и адгезионная способность улучшаются путем конструктивного изменения технологического участка вакуумной установки. Нередким случаем модернизируется система нагрева и устройства осаждения вакуумной камеры, причем для сохранения микроструктуры в объеме детали, должен производиться нагрев в диапазоне 200 – 300°С. Также для улучшения этих показателей проводится предварительная подготовка подложек, на которые будет осаждаться материал [2].

С таким параметром покрытия, как неравномерность, не все так просто, поскольку на него может влиять целый комплекс факторов, таких как: устройство осаждения, способ подачи в зону осаждения и метод осаждения материала.

В большинстве установок, предназначенных для большого числа покрытий, используется метод магнетронного осаждения материала. Это связано с тем, что он позволяет производить про-

цесс осаждения длительное время без замены материала осаждения на соответствующем устройстве.

Магнетронные устройства осаждения бывают различных типов. В комплексе со способом подачи тележек в технологическую зону они в значительной мере влияют как на неравномерность покрытия, так и качества покрытия в целом.

Зачастую магнетронные устройства обладают таким недостатком, как краевой эффект. Это вызвано тем, что магнитное поле магнетрона ослабляется у краев устройства и равномерно вдоль основной части устройства [3]. В свою очередь это приводит к тому, что покрытие получается неравномерным ближе к краям магнетронного устройства осаждения (см. рис. 2).

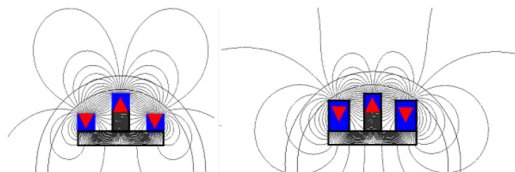


Рис. 2 – Краевой эффект сбалансированного и несбалансированного магнетронов

На практике для устранения краевых эффектов магнетронных устройств осаждения применяются соответствующие методики.

1. Устанавливают дополнительные экраны и магниты внутри технологической камеры напыления (см. рис. 3);
2. смещают все подложки к центру зоны напыления, в которой магнитное поле постоянно;
3. заменяют магнетроны на более сбалансированные, магнитное поле которых учитывает краевые эффекты;
4. анализируют влияние параметров подачи подложек в зону осаждения на качество получаемого покрытия.

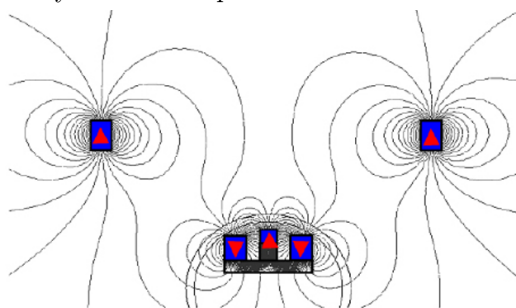


Рис. 3 – Устранение краевого эффекта путем введения дополнительных магнитов

IV. РАЦИОНАЛЬНОСТЬ МЕТОДОВ ПОВЫШЕНИЯ КАЧЕСТВА ПОКРЫТИЯ

Одним из самых приемлемых способов модернизации устройств распыления с экономической точки зрения – установка дополнительных экранов и магнитов внутри камеры напыления.

В этом случае можно добиться требуемого качества покрытия не меняя параметры транспортной системы, что позволит работать ей с максимальной производительностью и загрузкой оборудования.

Смещение всех подложек напыления ближе к центру напыляемой зоны будет являться экономически невыгодным решением, поскольку необходимо будет заменить все циркулирующие тележки в транспортной подсистеме вакуумной установки, поскольку даже одна тележка без изменений может привести к увеличению числа отбракованных покрытий.

Для туннельного вакуумного оборудования модернизация путем замены магнетронов на более сбалансированный также является экономически невыгодным решением, поскольку в зоне напыления находится сразу несколько устройств осаждения с различными материалами. От модернизации путем замены устройства напыления производитель покрытий на подложки потеряет вдвойне: убыток за счет покупки новых распылительных устройств и модернизация места их крепления к установке, а также за счет нереализованной выгоды с текущих магнетронных устройств.

Анализ влияния параметров транспортной системы на качество получаемого покрытия также является экономически выгодным способом устранения краевого эффекта. В этом случае подбираются такие параметры устройства осаждения и движения тележек, на которые краевой эффект не будет оказывать никакого влияния.

ЗАКЛЮЧЕНИЕ

Анализ качества покрытия, получаемого путем вакуумного осаждения материала в условиях массового осаждения на подложки, показал, что на получаемое покрытие оказывают влияние такие параметры как пористость, адгезионная способность и неравномерность. Причем последнему уделяется особое внимание в таких системах. Из рассмотренных методов устранения неравномерности покрытия наиболее экономически выгодными являются установка дополнительных экранов и магнитов, а также исследование параметров движения и устройств осаждения.

1. Физические основы вакуумной техники : учеб. пособие / А. Б. Беркин, А. И. Василевский. – Новосибирск : Изд-во НГТУ, 2014. – 84 с.
2. Джурицкий Д. В. Структура и свойства функционально-градиентных покрытий из аморфных и микрокристаллических сплавов, полученных методом сверхзвукового «холодного» газодинамического напыления для создания конкурентоспособных изделий – СПб, 2006. – 121 с.
3. Прикладная электроника [Электронный ресурс]. – Режим доступа: <http://apelvac.com/catalog/15/index.html>. – Дата доступа: 08.10.2020.

ИССЛЕДОВАНИЕ РОБАСТНОЙ УСТОЙЧИВОСТИ СИСТЕМЫ ПО ОДНОМУ ПАРАМЕТРУ В КЛАССЕ КАТАСТРОФЫ «ГИПЕРБОЛИЧЕСКАЯ ОМБИЛИКА»

Бейсенби М. А., Исатаева Г. С., Марков А. В.

Кафедра системного анализа и управления, Евразийский национальный университет им. Л.Н. Гумилева
Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники

Нур-Султан, Казахстан; Минск, Республика Беларусь

E-mail: beisenbi@mail.ru, gainel.issatayeva@gmail.com, markov@bsuir.by

ВВЕДЕНИЕ

Одним из ключевых направлений современной теории автоматического управления является анализ и синтез систем управления в условиях неопределенности. Это связано с различными факторами, такими как: неточное знание математической модели технологических процессов и технических объектов, упрощение описания модели, понижение степени сложности или пренебрежение имеющимися нелинейностями. Вследствие чего возникает потребность в создании таких автоматических систем, которые при изменении параметров объекта и влиянии внешних возмущений оставались бы не только в устойчивом состоянии, но и обеспечивали бы необходимое качество функционирования. Исследование и синтез таких систем проводятся в рамках теории робастного управления. Идея робастного проектирования заключается в нахождении таких установок управляющих параметров, при которых влияние шумовых факторов на выходные характеристики было бы минимальным [1]. В статье представлен универсальный подход к построению вектор-функции Ляпунова [2], основанный на геометрической интерпретации теоремы об асимптотической устойчивости прямого метода Ляпунова и понятиях устойчивости.

I. ПОСТАНОВКА ЗАДАЧИ

Будем рассматривать стационарную замкнутую систему управления с одним входом и одним выходом. Система описывается уравнением состояния:

$$\dot{x} = Ax + Bu \quad (1)$$

где $x(t) \in R^n$ — вектор состояния объекта; $u(t) \in R^1$ — скалярная функция управляющих воздействий; A — матрица объекта управления с неопределенными параметрами $n \times n$; B — матрица управления размерности $m \times 1$. Матрицы A и B имеют следующий вид:

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Закон управления $u(t)$ в замкнутом контуре задан в форме трехпараметрических

структурно-устойчивых отображений — катастрофы «гиперболическая омбилика» [3,4]:

$$u(x) = \frac{1}{b_n} (-x_1^3 - x_2^3 - k_1 x_1 x_2 + k_2 x_2 + k_3 x_1) \quad (2)$$

II. МЕТОДЫ ИССЛЕДОВАНИЯ

Систему управления (1) в развернутом виде можно представить как:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_{n-1} = x_n \\ \dot{x}_n = -x_1^3 - x_2^3 - k'_1 x_1 x_2 - (a_n - k_1)x_1 - \\ - (a_{n-1} - k_2)x_2 - a_{n-2}x_3 - \dots - a_2 x_{n-1} - a_1 x_n \end{cases} \quad (3)$$

Стационарные состояния системы определяются решением уравнения:

$$\begin{aligned} x_{2S} = 0, x_{3S} = 0, \dots, x_{n-1,S} = 0 \\ -x_{1S}^3 - x_{2S}^3 - k'_1 x_{1S} x_{2S} - (a_n - k_1)x_{1S} - \\ - (a_{n-1} - k_2)x_{2S} - a_{n-2}x_{3S} - \dots - a_1 x_{nS} = 0 \end{aligned} \quad (4)$$

Найдем стационарные состояния из уравнения (4):

$$x_{1S} = 0, x_{2S} = 0, \dots, x_{nS} = 0 \quad (5)$$

Другие стационарные состояния могут быть определены решением уравнений:

$$\begin{aligned} -x_{1S}^2 - a_n + k_1 = 0, x_{2S} = 0, \\ x_{3S} = 0, \dots, x_{nS} = 0 \end{aligned} \quad (6)$$

Уравнения (6) будут иметь мнимые решения при отрицательных $k_1 - a_n$ ($k_1 - a_n < 0$). Это не может соответствовать какой-либо физически возможной ситуации [5]. При $k_1 - a_n > 0$ уравнения (6) допускают следующие стационарные состояния:

$$\begin{cases} x_{1S}^1 = \sqrt{k_1 - a_n}, x_{2S} = 0, \dots, x_{nS} = 0, \\ x_{1S}^2 = -\sqrt{k_1 - a_n}, x_{2S} = 0, \dots, x_{nS} = 0 \end{cases} \quad (7)$$

Исследование устойчивости стационарных состояний (5) и (7) системы (1) будет проводиться методом вектор-функции Ляпунова [6].

Рассмотрим устойчивость стационарного состояния (5). Компоненты вектора градиента от

вектор-функции Ляпунова могут быть определены из (3):

$$\begin{cases} \frac{\partial V_1(x)}{\partial x_1} = 0, \frac{\partial V_1(x)}{\partial x_2} = -x_2, \frac{\partial V_1(x)}{\partial x_3} = 0, \dots, \frac{\partial V_1(x)}{\partial x_n} = 0, \\ \frac{\partial V_2(x)}{\partial x_1} = 0, \frac{\partial V_2(x)}{\partial x_2} = 0, \frac{\partial V_2(x)}{\partial x_3} = -x_3, \dots, \frac{\partial V_2(x)}{\partial x_n} = 0, \\ \dots \\ \frac{\partial V_n(x)}{\partial x_1} = x_1^3 + \frac{1}{2}k'_1x_1x_2 + (a_n - k_1)x_1, \\ \frac{\partial V_n(x)}{\partial x_2} = x_2^3 + \frac{1}{2}k'_1x_1x_2 + (a_{n-1} - k_2)x_2, \\ \frac{\partial V_n(x)}{\partial x_3} = a_{n-2}x_3, \dots, \frac{\partial V_n(x)}{\partial x_n} = a_1x_n \end{cases}$$

Разложение компонентов вектора скорости на координаты [6] найдем из уравнения состояния (3):

$$\begin{cases} \left(\frac{dx_1}{dt}\right)_{x_1} = 0, \left(\frac{dx_1}{dt}\right)_{x_2} = x_2, \\ \left(\frac{dx_1}{dt}\right)_{x_3} = 0, \dots, \left(\frac{dx_1}{dt}\right)_{x_n} = 0 \\ \left(\frac{dx_2}{dt}\right)_{x_1} = 0, \left(\frac{dx_2}{dt}\right)_{x_2} = 0, \\ \left(\frac{dx_2}{dt}\right)_{x_3} = x_3, \dots, \left(\frac{dx_2}{dt}\right)_{x_n} = 0 \\ \dots \\ \left(\frac{dx_{n-1}}{dt}\right)_{x_1} = 0, \left(\frac{dx_{n-1}}{dt}\right)_{x_2} = 0, \\ \left(\frac{dx_{n-1}}{dt}\right)_{x_3} = 0, \dots, \left(\frac{dx_{n-1}}{dt}\right)_{x_n} = x_n \\ \left(\frac{dx_n}{dt}\right)_{x_1} = -x_1^3 - \frac{1}{2}k'_1x_1x_2 - (a_n - k_1)x_1, \\ \left(\frac{dx_n}{dt}\right)_{x_2} = -x_2^3 - \frac{1}{2}k'_1x_1x_2 - (a_{n-1} - k_2)x_2, \\ \left(\frac{dx_n}{dt}\right)_{x_3} = -a_{n-2}x_3, \dots, \left(\frac{dx_n}{dt}\right)_{x_n} = -a_1x_n \end{cases}$$

Полная производная по времени от вектор-функции Ляпунова определяется как скалярное произведение:

$$\begin{aligned} \frac{dV(x)}{dt} &= \sum_{i=1}^n \left(\sum_{j=1}^n \frac{\partial V_i(x)}{\partial x_j} \left(\frac{dx_j}{dt}\right)_{x_j} \right) = \\ &= -x_2^2 - x_3^2 - \dots - x_n^2 - \\ &- \left(x_1^3 + \frac{1}{2}k'_1x_1x_2 + (a_n - k_1)x_1\right)^2 - \\ &- \left(x_2^3 + \frac{1}{2}k'_1x_1x_2 + (a_{n-1} - k_2)x_2\right)^2 - \\ &- (a_{n-2}x_3)^2 - \dots - (a_1x_n)^2 \end{aligned} \quad (8)$$

Как мы можем увидеть из (8), полная производная от вектор-функции Ляпунова всегда отрицательно-определенная функция, т.е. всегда выполняется достаточное условие асимптотической устойчивости системы.

Построим функцию Ляпунова в следующем виде по градиенту этой функции:

$$\begin{aligned} V(x) &= \frac{1}{4}x_1^4 + \frac{1}{4}x_2^4 + \frac{1}{4}k'_1x_1^2x_2^2 + \frac{1}{2}(a_n - k_1 - 1)x_1^2 + \\ &+ \frac{1}{2}(a_{n-1} - k_2 - 1)x_2^2 + \\ &+ \frac{1}{2}(a_{n-2} - 1)x_3^2 + \dots + \frac{1}{2}(a_1 - 1)x_n^2 \end{aligned} \quad (9)$$

Если $k'_1 > 0$, то положительная определенность функции Ляпунова (9) определяется условиями:

$$\begin{cases} a_n - k_1 - 1 > 0 \\ a_{n-1} - k_2 - 1 > 0 \\ a_{n-2} - 1 > 0 \\ \vdots \\ a_1 - 1 > 0 \end{cases} \quad (10)$$

Таким образом, стационарное состояние (5) системы (3) будет асимптотически устойчивым при $k'_1 > 0$, если условия (10) выполняются.

ЗАКЛЮЧЕНИЕ

В статье был предложен новый подход к исследованию робастной устойчивости системы управления, полученный из геометрической интерпретации теоремы А.М. Ляпунова. Функция Ляпунова синтезируется в форме вектор-функции, антиградиент которой задается компонентами вектора скорости (правой частью уравнения состояния) системы в форме тензора. Условия устойчивости получаются из положительной определенности вектор-функции Ляпунова в форме системы неравенств по неопределенным параметрам объектов управления и заданным параметрам регулятора. В конечном итоге, исследования показывают, что система управления с повышенным потенциалом робастной устойчивости для линейных объектов с одним входом и одним выходом асимптотически устойчива в положительной и отрицательно определенных областях стационарных состояний неопределенных параметров объекта управления. Стационарные состояния системы определяются путем синтеза закона управления в классе катастрофы «гиперболическая омбилика».

1. Ковриго, Ю. М. Синтез робастного регулятора для объектов с изменяющимися параметрами /Ю. М. Ковриго, Т. Г. Баган // Доклады БГУИР – 2015. – № 2. – С. 168–171.
2. Бейсенби, М. А. Построение функции Ляпунова в исследовании робастной устойчивости линейных систем /М. А. Бейсенби, Ж. Ж. Ермекбаева // Вестник КазНТУ им. К.Сатпаева. – Алматы, 2013. – № 1. – С. 315–320.
3. Бейсенби, М. А. Системы управления с повышенным потенциалом робастной устойчивости /М. А. Бейсенби, Б. А. Ержанов. – Астана, 2002. – 164 с.
4. Бейсенби, М. А. Методы повышения эффективности робастной устойчивости систем управления. – Астана, 2011. – 352 с.
5. Gregoire Nicolis, Prigogine Иуа. Exploring Complexity: An introduction. – New York: W.H. Freeman and Company, 1989.
6. Бейсенби, М. А. Исследование робастной устойчивости систем автоматического управления методом функции А.М. Ляпунова. – Астана: DR-Project Publ., 2015.

ТЕХНОЛОГИЯ ETHERNET OVER USB КАК ОСНОВА ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА НА RASPBERRY PI

Калоша С. О., Нестеренков С. Н.

Кафедра программного обеспечения информационных технологий, факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: swetofdark@gmail.com, nsn@bsuir.by

Кластер из нескольких вычислительных устройств Raspberry PI является весьма популярным решением в последнее время. Зачастую его реализация подразумевает использование «стандартных» подходов с дополнительным сетевым оборудованием, беспроводными сетями и написанием низкоуровневых программных пакетов. В данной статье предлагается альтернативный подход по реализации небольших Master-Slave кластеров, основанный на использовании Ethernet over USB технологии и реализуемый простым конфигурированием операционной системы.

ВВЕДЕНИЕ

В рамках работы над магистерской диссертацией, целью которой является исследование моделей машинного обучения для классификации объектов реального мира, передо мной была поставлена задача проектирования устройства для сканирования твердых тел небольшого размера и массы. Данное устройство включило в себя несколько фотокамер. А полученным на них изображениям требовалась обработка. Для сохранения возможности использовать различные алгоритмы обработки изображений, а также высокой степени параллелизации таких вычислений было принято решение управлять каждой камерой и ее выводом отдельным вычислительным устройством. При этом, одно из таких вычислительных устройств также должно было координировать работу всех устройств между собой, предоставлять интерфейс взаимодействия с пользователем и обрабатывать данные с других датчиков (см. рис. 1).

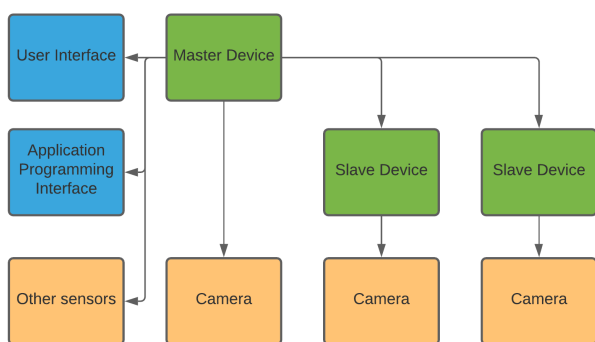


Рис. 1 – Схема устройства для сканирования объектов

Исходя из изложенных условий, было принято условие использовать микрокомпьютеры семейства Raspberry PI [1]. Эти микрокомпьютеры обладают достаточной вычислительной мощностью для обработки изображений, имеют возможность подключать камеры через отдельно выделенный интерфейс, а также позволяют подключать различные цифровые датчики. В каче-

стве координирующего устройства (Master) был выбран Raspberry PI 4, в качестве остальных устройств (Slave) – Raspberry PI Zero v1.2. После чего было необходимо решить каким образом настроить взаимодействие данных устройств между собой.

I. ВАРИАНТЫ СОЕДИНЕНИЙ МЕЖДУ RASPBERRY PI

Существует несколько способов настроить обмен данными между несколькими Raspberry PI. Ниже приведены рассмотренные некоторые из таких способов:

- настройка последовательных соединений SPI, UART [2] через GPIO [3];
- проводное ethernet-соединение [4];
- беспроводное Wi-Fi соединение [4];
- Ethernet-соединение через USB.

Настройка GPIO портов для последовательных соединений требует низкоуровневого программирования, в следствие чего такие соединения будут более сложными в поддержке. Также, Raspberry обладают ограниченным набором GPIO-портов и не каждые из них могут быть использованы для SPI или UART соединений, более того, использование этих портов сильно уменьшает количество датчиков, подключаемых к микрокомпьютеру.

Соединение Ethernet-проводом для более чем двух выбранных устройств невозможно без дополнительного сетевого оборудования, в силу наличия только одного Ethernet-разъема, а также ограниченный в использовании Ethernet одновременно с модулем камеры на Raspberry Pi Zero.

Беспроводное Wi-Fi соединение позволяет объединить более двух устройств в одну сеть, однако это ограничивает возможности по подключению Master к домашним беспроводным сетям, а также делает возможным сетевое обнаружение всех вычислительных узлов конечного устройства, что нарушает принцип «черного ящика».

В силу вышеуказанных недостатков было принято решение настроить соединение, осно-

ванное на Ethernet over USB технологии. Такой подход позволит подключить к Raspberry PI 4 до четырех устройств, и настроить взаимодействие по принципу Master-Slave между ними по сетевым протоколам прикладного уровня модели ISO/OSI, таким как HTTP.

II. НАСТРОЙКА ETHERNET OVER USB

Для настройки Ethernet over USB в первую очередь необходимо записать оригинальный образ операционной системы на SD-карту, используемую в устройстве. Образ исходной операционной системы может быть записан с помощью утилиты Raspberry PI Imager. Далее устройство должно быть переведено в режим OTG-гаджета включением `dwc2` оверлея дерева устройств. Для этого в конец файла `config.txt` необходимо добавить строку `dtoverlay=dwc2`.

Следующим шагом является установка параметров запуска ОС в файле `cmdline.txt`. Здесь необходимо указать загружаемые модули `modules-load=dwc2,g_ether` сразу после параметра `rootwait`, которые переопределяют USB-порты как Ethernet-порты. [5]

Для простой проверки, что устройство функционирует в режиме Ethernet over USB, нужно создать `ssh` файл в образе операционной системы, что включит `ssh` модуль ОС. И после ее установки достаточно подключить Raspberry PI к компьютеру через USB, и установить SSH соединение по сетевому имени по умолчанию `raspberrypi.local`.

III. НАСТРОЙКА IP-АДРЕСАЦИИ

После конфигурации Ethernet over USB для всех устройств, каждое подключение к Master через USB Slave ведет к созданию сетевого интерфейса `usb*`, например `usb0`. Однако получить IP-адрес подключенного к этому интерфейсу узла невозможно без использования специальных модулей сетевого обнаружения. В случае, когда топология получившейся сети заранее известна и проста, данная проблема решается установкой статических IP-адресов для каждого подключаемого устройства.

Большинство существующих в данный момент руководств предлагают решать данную задачу редактированием файла `/etc/network/interfaces`. Однако такие изменения в Raspberry PI OS приведут к нарушению работы других сетевых интерфейсов, например `wlan`.

Поэтому нужно изменить `dhcpcd` настройки ОС в файлах `/etc/dhpcd.conf` Master и Slave устройств. Так, например для Master `usb0` интерфейс может быть настроен следующим образом:

```
interface usb0
static ip_address=192.168.1.1
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Для Slave-устройства, присоединенного через сконфигурированный интерфейс, должны использоваться те же настройки, за исключением `ip_address`.

IV. ПРАВИЛА СОЗДАНИЯ СЕТЕВЫХ ИНТЕРФЕЙСОВ

В Raspberry PI OS `usb*` сетевые интерфейсы не привязаны к конкретному физическому разъему, поэтому выполненная ранее настройка IP-адресов может давать сбои в случае, когда к Master одновременно подключены несколько Slave-устройств. Решить данную проблему можно созданием привязки имени сетевого интерфейса к USB-разъему.

В философии UNIX-систем всё, в том числе и сетевые интерфейсы, представлено файлом [6]. Файлы, представляющие устройства, хранятся в директории `/dev`, и управляются утилитой `UDEV`. Данная утилита использует набор правил, хранящихся в директории `/etc/udev/rules.d`. Добавление нового правила в эту директорию позволит переименовать устройство или изменить права доступа к нему. Поэтому добавление файла со следующим правилом позволит привязать интерфейс `usb0` к конкретному USB-разъему Raspberry PI 4 В:

```
SUBSYSTEM=='net', SUBSYSTEMS=='usb',
KERNELS=='1-1.3:*', NAME='usb0'
```

ЗАКЛЮЧЕНИЕ

Изложенный выше подход по настройке сетевых соединений между несколькими Raspberry PI через USB позволил быстро получить систему из нескольких взаимодействующих вычислительных узлов без дополнительных программных и аппаратных затрат. Аспекты конфигурирования, описанные в данной статье могут оказаться полезными для настройки подобных соединений. Стоит отметить, что Ethernet over USB не является единственно верным вариантом, поэтому для каждой конкретной задачи стоит начать со сравнения альтернатив, как, например, было сделано в первом разделе данной статьи.

1. Raspberry Pi Documentation [Электронный ресурс] / Raspberry Pi Foundation. – Режим доступа: <https://www.raspberrypi.org/documentation/>. – Дата доступа: 10.10.2020.
2. Дэвид М. Харрис, Сара Л. Харрис. Цифровая схемотехника и архитектура компьютера. / Imagination Technologies. – М.: ДМК Пресс, 2018. – 792 с.
3. Монк, Саймон. Raspberry Pi. Сборник рецептов: решение программных и аппаратных задач, 2-е изд. / СПб.: ООО «Диалектика», 2019. – 528 с.
4. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. / СПб.: Питер, 2017. – 960 с.
5. The GNU/Linux USB driver [Электронный ресурс] / Linux-USB Org. – Режим доступа: <http://www.linux-usb.org/usbnet/>. – Дата доступа: 10.10.2020.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. / СПб.: Питер, 2019. – 1120 с.

ОПРЕДЕЛЕНИЕ НЕОБХОДИМЫХ ЭЛЕМЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА И ПОСЛЕДОВАТЕЛЬНОСТИ ИХ РАЗМЕЩЕНИЯ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ

Капцевич О. А., Рабченко Д. И.
ООО «ИнноТех Солошнс»

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: vip.alekh@mail.ru

Представлены аналитические выражения, позволяющие определить необходимые элементы пользовательского интерфейса и обосновать последовательность их размещения в автоматизированной системе управления для осуществления эффективного взаимодействия человека-оператора с автоматизированным рабочим местом.

ВВЕДЕНИЕ

Автоматизированная система управления (АСУ) сложной системой содержит разнородные элементы, предназначенные для решения задач обработки информационных потоков. Одним из таких элементов является автоматизированное рабочее место (АРМ) человека-оператора, в состав которого входит пользовательский интерфейс (ПИ). Проектирование и совершенствование ПИ АРМ для управления сложными системами является трудоемкой задачей по ряду причин, в том числе формирования объема информации, циркулирующей в АСУ, исходя из необходимости обработки значительных массивов данных об объектах управления, внешней среде и условиях организации управления, причем в ограниченное время. Применяемый при разработке современных ПИ АРМ АСУ математический аппарат отличается наличием нерешенных задач по формализации отдельных составляющих этапов переработки человеком-оператором информации, в том числе его взаимодействием с элементами ПИ [1, 2]. Актуальной научно-практической задачей является разработка аналитических выражений, предназначенных для определения состава и структуры ПИ на основании априорных данных, с учетом последовательности и структуры диалога, «сценария» и «действий сторон», расположения и характера смены информации.

I. ОПРЕДЕЛЕНИЕ НЕОБХОДИМЫХ ЭЛЕМЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

В идеальном случае каждому варианту управления $u_i \in U$ ставится в соответствие единственное состояние внешней среды $r_i \in R$. В реальных системах каждому допустимому варианту u_i управления могут соответствовать различные r_i и соответствующие результаты управления u_{ij} . Под результатом u_{ij} подразумевается оценка соответствующему варианту управления u_i при усло-

вии r_j . Семейство результатов, характеризующее множество стратегий (вариантов) управления U для множества состояний внешней среды R как правило представляется в виде матрицы решений [3, 4]. Чтобы найти один из выгодных вариантов управления даже в том случае, когда каким-то вариантам u_i могут соответствовать различные состояния внешней среды r_j , вводят определенную оценочную (целевую) функцию. Оптимальная с точки зрения гарантированного результата реализация управления находится при помощи выражения:

$$u^{opt} = \{u_i^{opt} | u_i^{opt} \in U \wedge u_i^{opt} = \max_i(\min_j \tau_{ij})\}, \quad (1)$$

где τ_{ij} - длительность реализации управления u_i при условиях r_j .

Выбор наилучшей реализации управления (и необходимого набора элементов ПИ) в соответствии с критерием Вальда [5] не является однозначным, поскольку результат может достигаться на всем множестве результатов многократно [6]. На практике выбирается одна из нескольких лучших реализаций. Для выбранной реализации определяются наиболее стереотипные по отношению к предметной области исследований варианты (наборы элементов) интерфейса. Каждый вариант предполагает количественную оценку при помощи аналитических выражений.

II. ОПРЕДЕЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Усредненная частота использования j -го элемента ПИ при m -ом обращении человека-оператора определяется как:

$$P_{mj} = \frac{1}{N_{alg}} \cdot \sum_{n=1}^{N_{alg}} B_{mjn} \quad (2)$$

где N_{alg} - количество возможных алгоритмов действий человека-оператора, $B_{mjn} = 1$ - если

в n -ом алгоритме при m -ом обращении присутствует элемент j , а в остальных случаях $V_{mjn} = 0$

Первым в очереди для размещения элементом ПИ выбирается элемент для которого величина P_{mj} является максимальной. В случае наличия нескольких подобных элементов, в качестве первого элемента выбирается любой из претендентов.

Усредненная частота следования элемента ϕ за элементом ξ , в рамках выполнения человеком-оператором n -ого алгоритма управления, определяется выражением:

$$S_{\xi\phi} = \frac{1}{N_{alg}} \cdot \sum_{n=1}^{N_{alg}} Y_{\xi\phi n} \quad (3)$$

где $Y_{\xi\phi n}$ – количество случаев, когда в n -м алгоритме элемент ϕ следует за элементом ξ .

Для выбора второго в очереди элемента для размещения рассчитывается ряд величин

$$o_j = P_2 + \frac{1}{2} \cdot S_{M_1 j}$$

для всех значений j (всех элементов), за исключением первого элемента. Находится элемент (элементы) для которого величина o_j максимальна. Данный элемент (элементы) будет являться вторым (M_2) элементом для размещения.

В общем случае, для выбора очередного M_β элемента ПИ для размещения рассчитывается ряд величин:

$$o_j = P_{\beta j} + \frac{1}{2} \cdot S_{M_{(\beta-1)j}} \quad (4)$$

для всех значений j , не равных $M_1, M_2, M_{(\beta-1)}$ (элементов ПИ которые уже размещены). Множитель при величине $S_{M_{(\beta-1)j}}$ характеризует ее меньший, по сравнению с величиной $P_{\beta j}$, вклад в оценку элемента ПИ. Элемент для которого величина o_j окажется максимальной будет являться вторым элементом для размещения. Третий и последующие элементы для размещения выбираются аналогичным образом.

Для оценки различных вариантов (в случае наличия нескольких претендентов на роль первого элемента) размещения элементов в ПИ необходимо определить для каждого из них следующий показатель:

$$\Delta = \frac{2}{n_{max}^3 + n_{max}} \sum_{n=1}^{n_{max}} (\Upsilon_n - 1)(n_{max} + 1 - n) \quad (5)$$

Показатель Δ представляет собой усредненный по длине очереди, взвешенный по убыванию

предпочтения нахождения на позициях в очереди (согласно схеме весов Фишберна [7]) и количеству элементов на позициях коэффициент. Наилучшим является вариант размещения элементов ПИ на АРМ, для которого указанный показатель минимален.

III. ВЫВОДЫ

Предлагаемые аналитические выражения (1)–(5) позволяют определять необходимые элементы ПИ, а также последовательность их размещения в системе управления для осуществления взаимодействия человека-оператора с АРМ в АСУ.

Использование априорной информации позволит значительно снизить объем трудоемких экспериментальных исследований деятельности человека-оператора, связанных с многовариантностью его действий в различных условиях обстановки, и требующих существенных затрат различного характера. Использование предлагаемых аналитических выражений позволит эффективнее по сравнению с существующими математическими аппаратами учитывать человеческий фактор, оказывающий существенное влияние на процесс управления сложной системой в условиях дефицита времени и высокой стоимости ошибки

1. Капцевич, О. А. Временная составляющая аналитической модели действий диспетчера управления воздушным движением / О. А. Капцевич, А. В. Дубовский, Д. И. Рабченко // Доклады БГУИР. – 2019. – № 5. – С. 79–85.
2. Круликов, С. В. Аналитическая модель взаимодействия оператора с информационной моделью автоматизированного рабочего места в условиях динамично меняющейся обстановки / С. В. Круликов, Д. И. Рабченко // Информационно-измерительные и управляющие системы. – 2016. – № 3. – С. 37–44.
3. Системное проектирование средств отображения информации в автоматизированных системах управления / Б. М. Герасимов. – Киев: КВИРТУ ПВО, 1983. – 348 с.
4. Теория выбора и принятия решений: учебник / И. М. Макаров, Т. М. Виноградская. – М.: Наука, 1982. – 328 с.
5. Концептуальные основы управления в условиях неопределенности / Ю. С. Мануйлов. – СПб.: ВКА им. А.Ф. Можайского, 2008. – 121 с.
6. Сложные системы : учеб. пособие / А. С. Шаракшанэ. – М.: Высш. школа, 1977. – 247с.
7. Ремесник, Е. С. Применение последовательностей Фишберна в когнитивных гибридных системах поддержки принятия решений / Е. С. Ремесник // Теория и практика общественного развития. – 2020. – № 6. – С. 38–42.

ОРТОГОНАЛИЗАЦИЯ СИСТЕМЫ ДНФ БУЛЕВЫХ ФУНКЦИЙ

Кардаш С. Н.

Объединённый институт проблем информатики Национальной академии наук Беларуси
Минск, Республика Беларусь
E-mail: gold@newman.bas-net.by

Предлагается новый алгоритм ортогонализации системы дизъюнктивных нормальных форм булевых функций.

I. ВВЕДЕНИЕ

Для решения многих задач синтеза, диагностики и анализа надежности технических систем используется представление булевых функций в виде дизъюнктивных нормальных форм (ДНФ). Часто бывает полезно иметь такие ДНФ, в которых все входящие в них элементарные конъюнкции взаимно ортогональны. Для получения таких ДНФ необходимо проводить ортогонализацию исходных систем ДНФ. В работе [1] даны как необходимые понятия, так и некоторые идеи, способствующие решению этой задачи. В настоящей работе приводится алгоритм, на основе которого разработана компьютерная программа, решающая задачу ортогонализации системы ДНФ и некоторые результаты ее экспериментального исследования.

В случае небольшого числа переменных задачу ортогонализации ДНФ можно решить, разложив дизъюнктивно каждую элементарную конъюнкцию по всем отсутствующим в ней переменным, и после приведения подобных получить в результате совершенную ДНФ. Однако такой способ может оказаться неприемлем, когда переменных много. В частности, для системы ДНФ, зависящих от n переменных, число конъюнкций в ортогонализированной системе может достигать 2^n .

Предлагаемый ниже алгоритм основан на идее дизъюнктивного разложения элементарной конъюнкции на серию других конъюнкций, каждая из которых либо ортогональна всем конъюнкциям из некоторой совокупности, либо поглощается одной из них. В последнем случае поглощаемая конъюнкция удаляется из решения, а число остающихся конъюнкций по возможности минимизируется – это сокращает последующие вычисления.

II. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ.

Представим исходную систему ДНФ в матричном виде парой булевых матриц – U (троичной) и S (булевой). Столбцы матрицы U соответствуют аргументам системы, а строки задают элементарные конъюнкции. Столбцы матрицы S соответствуют функциям системы, а единичные значения элементов в матрице S отмеча-

ют вхождения соответствующих конъюнкций в ДНФ функций системы.

Строки троичной матрицы задаются троичными векторами, а строки булевой матрицы – булевыми. Строки u и s представляют матрицы U и S соответственно. Троичный вектор u представляется парой булевых векторов U_0 и U_1 . Троичный вектор w представляется парой булевых векторов W_0 и W_1 . Булевы вектора g и s представляются булевыми векторами G_1 и S_1 соответственно.

Определим следующие бинарные отношения на множестве векторов одинаковой размерности.

Ортогональность. Троичные векторы u и v ортогональны по i -й компоненте, если u_i и v_i только если i -я компонента имеет значение 0 в одном из этих векторов и 1 – в другом. Троичные векторы ортогональны, если они ортогональны хотя бы по одной компоненте.

Поглощение. Троичный вектор w поглощает троичный вектор u , тогда и только тогда, когда все компоненты вектора w , значения которых отличны от «-», совпадают с одноименными компонентами вектора u .

Булев вектор g поглощает булев вектор s , если всем единичным компонентам вектора s соответствуют единичные компоненты вектора g .

Склеивание булевых векторов. Два булевых вектора можно заменить одним вектором, у которого значения компонент определяются следующим образом. Компонента, в которой соответствующая компонента хотя бы одного исходного вектора имела единичное значение, приобретает значение «1». Значения остальных компонент получают значение «0».

Постановка задачи. Пусть задана система ДНФ булевых функций. Требуется построить эквивалентную ортогонализированную систему, содержащую минимальное число элементарных конъюнкций.

Для решения этой задачи предлагается эвристический алгоритм.

III. ОПИСАНИЕ АЛГОРИТМА

Результат ортогонализации представляется парой матриц – W (троичной) и G (булевой).

Строки w и g представляют матрицы W и G соответственно.

Результат разложения двух строк представляется тройной матрицей V , каждая строка v которой представляется парой булевых векторов V_0 и V_1 .

Алгоритм. Матрица W полагается пустой (не содержащей строк).

1. В матрицу W заносится первая строка u и матрицы U , a в матрицу G заносится первая строка s матрицы S .

2. Из матрицы U выбирается очередная строка u . Если все строки матрицы U просмотрены – переход на п. 3.

2.1. Выбирается очередная строка w матрицы W . Если все строки матрицы W просмотрены – переход на п. 3. Иначе – строки u и w сравниваются.

Если u и w ортогональны – переход на п.2.1.

Если строки u и w совпадают, то строки s и g склеиваются, и результат склеивания заменяет строку g в матрице G , переход на п.3.

Если w поглощает u , а g поглощает s , то переход на п.3.

Для строк u и w строится разложение – тройная матрица V . Алгоритм построения разложения описан в [1].

2.2. Первые $n-1$ строк матрицы V переносятся в матрицу W , а соответствующие $n-1$ строк матрицы G , являются копией строки S_1 . Последняя строка матрицы V добавляется в матрицу W , строки s и g склеиваются, и результат склеивания добавляется в матрицу G . Переход на п.2.

3. Если на шаге 2 были склеивания или поглощения - переход на п.2. Строка u добавляется в матрицу W , а строка s – в матрицу G . Переход на п.2.

4. Если на шаге 2 не было разложений - переход на п.5. В матрицу U в обратном порядке переносятся строки матрицы W . Переход на п.1.

5. Конец.

IV. ОРГАНИЗАЦИЯ ЭКСПЕРИМЕНТА

Для проверки эффективности предложенного алгоритма был проведен вычислительный эксперимент.

Всего для каждого примера рассматривалось четыре варианта ортогонализации, представленные в таблице столбцами А, Б, В, Г. При первом (столбец А) – использовалась программа, реализующая алгоритм, предложенный в [1], в остальных случаях – описанный выше. Причем, во втором случае (столбец Б) производилось предварительное упорядочивание строк матриц U и S по возрастанию весов строк матрицы U , в третьем (столбец В) – по убыванию, а в четвертом (столбец Г) – упорядочивание не производилось.

Примеры матричных описаний систем полностью определенных булевых функций были взяты из набора промышленных тестовых примеров, входящих в библиотеку примеров Berkeley PLA Test Set [2].

V. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ

Результаты экспериментального исследования представлены в таблице 1. Здесь: n – число переменных, m – число функций, k – число элементарных конъюнкций исходной системы ДНФ булевых функций. Результатом является число элементарных конъюнкций в ортогонализованной системе.

ЗАКЛЮЧЕНИЕ

Эксперимент показал, что для исследованного множества примеров использование нового алгоритма во всех случаях обеспечивало нахождение лучшего решения, при этом программа работала быстрее.

СПИСОК ЛИТЕРАТУРЫ

1. Поттосин, Ю.В., Шестаков Е.А. Ортогонализация системы полностью определенных булевых функций / Ю.В.Поттосин, Е.А.Шестаков / Логическое проектирование, Вып.5. – Минск: Институт технической Кибернетики НАН Беларуси, 2000 г. – С. 107–115.
2. Berkeley PLA test set [Electronic resource]. – Mode of access: <http://www1.cs.columbia.edu/cs4861/sis/espresso-examples/ex/>. – Date of access: 10.10.2011.

Таблица 1 – Результаты эксперимента

Имя	n	k	m	А	Б	В	Г
Alu1	12	8	19	2531	1683	1598	1924
B2	16	17	110	249	175	206	185
Mp2d	14	14	123	813	580	541	537
newtpla	15	5	23	127	86	108	93
X6dn	39	5	121	560	364	368	380
sex	9	14	23	185	209	210	165
In2	19	10	137	5369	955	1316	1772
tial	14	8	640				10358

СТАНДАРТЫ ТОКЕНОВ СЕТИ ETHEREUM И ИХ ОСОБЕННОСТИ В РЕАЛИЗАЦИИ РАБОТЫ СМАРТ-КОНТРАКТОВ

Набешко Г. А., Захарьев В. А.

Кафедра систем управления, кафедра защиты информации, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: gnabeshko@gmail.com, zahariev@bsuir.by

Данная статья посвящена исследованию стандартов токенов сети Ethereum. В ней представлена информация о наиболее популярных видах токенов, подробно рассмотрены ключевые особенности в реализации работы смарт-контрактов для каждого вида токена, приведен сравнительный анализ.

ВВЕДЕНИЕ

Реестр блоков транзакций (блокчейн) — выстроенная по определённым правилам непрерывная последовательная цепочка блоков, содержащих информацию. Чаще всего копии цепочек блоков хранятся на множестве разных компьютеров независимо друг от друга.

Блокчейн — это, по существу, распределённая база данных или публичная книга всех транзакций или цифровых событий, которые были выполнены и распределены между участвующими сторонами в одноранговых сетях.

Токен — это единица учёта, не являющаяся криптовалютой, предназначенная для представления цифрового баланса в некотором активе, иными словами — выполняющая функцию «заменителя ценных бумаг» в цифровом мире.

Технологически токены реализуются на базе некоторой блокчейн платформы, обеспечивающей возможность записи в распределённый реестр блоков балансов токенов и транзакций. В настоящее время основной платформой для создания токенов является Ethereum.

I. СМАРТ-КОНТРАКТЫ

Смарт-контракт (англ. Smart contract — умный контракт) — компьютерный алгоритм, предназначенный для формирования, контроля и предоставления информации о владении чем-либо. Чаще всего речь идет о применении технологии блокчейна. В более узком смысле под смарт-контрактом понимается набор функций и данных (текущее состояние), находящихся по определённому адресу в блокчейне.

Базовые принципы смарт-контрактов уже были реализованы в блокчейне Биткойна, но их возможности были сильно ограничены и не подходили для того, чтобы создавать на их основе токены для отдельных распределённых приложений. Именно по этой причине можно считать, что Ethereum является первой платформой, которая полноценно развила идею смарт-контрактов и внедрила ее в работу.

Основная идея работы смарт-контракта заключается в следующем: когда вы проводите

транзакцию, отправляя токены на адрес получателя, на самом деле отправляется инструкция в смарт-контракт о том, что у данного количества монет сменился адрес владельца. Именно в результате данного действия у отправителя токены пропадают, а у получателя появляются.

Внутри смарт-контракта можно прописать уникальную механику токена. Он будет работать, но другие разработчики (и смарт-контракты) не смогут обращаться к нему с помощью универсальной логики.

На помощь приходят стандарты. Т.к. Ethereum является open-source проектом (кстати, ERC — это Ethereum Request for Comments), логично, что новый стандарт токена может предложить любой пользователь. Если стандарт решает какую-то важную проблему, то он может стать официальным стандартом Ethereum. Все ERC формализуются в виде EIP (Ethereum Improvement Proposal/предложения по улучшению сети Ethereum).

В настоящее время на веб-странице Ethereum EIP имеется более 150 EIP, при этом часто появляются и новые. Мы рассмотрим 5 самых популярных EIP:

- ERC20 — Смарт-контракт ICO;
- ERC223 — Предотвращает отправку токенов некорректному адресату;
- ERC721 — Не взаимозаменяемые токены;
- ERC777 — Новый расширенный стандарт токена;
- ERC1155 — Стандарт мульти-токена.

II. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СТАНДАРТОВ ЦИФРОВЫХ ТОКЕНОВ

1. ERC20 Standard — смарт-контракт ICO
Стандарт ERC20 является самым популярным для ICO. Стандарт ERC-20 определяет набор правил, которые должны быть соблюдены для того, чтобы токен был принят и имел возможность взаимодействовать с другими токенами в сети.

Сценарии использования токенов ERC-20 самые различные. Например, они могут выступать как акции проекта, сертификаты, подтвер-

ждающие владение активами, очки в программах лояльности или как криптовалюта.

Преимущества использования ERC-20:

- Упрощается листинг на криптовалютных биржах;
- Улучшается взаимодействие с разными сетями;
- Аудит смарт-контрактов становится более предсказуем и стандартизирован;
- Становится легче создавать и поддерживать пользовательские инструменты и кошельки, вокруг экосистемы Ethereum.

Стандарт ERC-20 предусматривает шесть обязательных функций для любого смарт-контракта.

Функции:

Total Supply: Общая эмиссия токенов.

Balance: Изначальное количество токенов.

Allowance: Сумма, которую разрешено снимать с владельца.

Transfer: Передача токена по указанному адресу.

Approve: Позволяет снимать определенное количество токенов с договора.

Transfer From: Передача токенов с источника на адрес.

2. ERC223 Standard – Предотвращает отправку токенов некорректному адресату. ERC223 был предложен Daxagan в декабре 2017 года. Стандарт ERC20 страдает от невозможности обработки входящих транзакций по контракту от получателя. Большая проблема заключается в том, как не предназначенные для адресата токены могут быть отправлены по контракту. Эти средства могут потеряться.

Стандарт ERC223 представляет новую функцию отмены транзакций, которой нет в существующем стандарте ERC20, чтобы предотвратить случайную передачу токенов. Это оптимальное решение, чтобы токены не потерялись в сети Ethereum, что очень важно.

3. ERC721 Standard – Не взаимозаменяемые (non-fungible) токены ERC721 Standard – один из более перспективных ERC-стандартов, который применяется для активно развивающихся не взаимозаменяемых токенов. Не взаимозаменяемыми называются токены, или активы, которые нельзя обменять на другой такой же токен, или актив, в отличие от стандартных эфириума и биткойна. Самые известные из таких токенов – это CryptoKitties.

Следующий стандарт позволяет реализовать стандартный API для не взаимозаменяемых токенов (NFT) в смарт-контрактах. Этот стандарт обеспечивает базовый функционал для отслеживания и передачи NFT. Они могут представлять права собственности на цифровые или физические активы.

4. ERC777 – новый расширенный стандарт токена. Этот стандарт определяет новый способ взаимодействия с контрактами по токенам, оставаясь полностью совместимым с ERC20.

Особенности:

- Стандарт позволяет использовать одну транзакцию для работы со смарт-контрактами вместо двух при обмене одного токена на другой;
- Соблюдение процедур AML/KYC;
- Возможность отклонить передачу и получение токена;
- Держатели токенов могут разрешать/запрещать передачу токенов другими операторами от их лица;
- Дополнительное поле для свободных сообщений-комментариев.

5. ERC1155 – Стандарт мульти-токена

Идею этого стандарта предложили разработчики проекта Enjin в 2018. Enjin – проект, стремящийся к упрощению создания игр на Ethereum.

Особенности:

- Позволяет выпускать несколько токенов в одном контракте;
- Токены в одном контракте могут быть fungible и non-fungible одновременно;
- Поддерживает атомарные свопы;
- Поддерживает «batch» транзакции;
- Не для всех транзакций нужно ждать окончания блока.

III. СПИСОК ЛИТЕРАТУРЫ

1. Ethereum Improvement Proposals [Электронный ресурс]. – Режим доступа: <https://eips.ethereum.org/erc>.
2. Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki>.
3. GitHub. Ethereum Improvement Proposals [Электронный ресурс]. – Режим доступа: <https://github.com/ethereum/EIPs>.
4. Introducing Ethereum and Solidity / Dannen, Chris // Apress – 2017.

АВТОМАТИЗАЦИЯ РАЗРАБОТКИ ПОРТАЛОВ С МНОГОКРАТНО ИСПОЛЬЗУЕМЫМИ КОМПОНЕНТАМИ

Навросяк В. Г.

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники Минск, Республика Беларусь
Минск, Республика Беларусь
E-mail: navrosjuk@gmail.com

Данная работа рассматривает принципы разработки веб-порталов. Исследуются возможности применения многократно используемых компонент, их комбинация, а также использование контент-менеджера.

ВВЕДЕНИЕ

В наше время огромную долю IT рынка занимает разработка веб-сайтов. Веб-порталы с многократно используемыми компонентами позволяют упростить процесс создания веб-сайтов, а также реализовать дальнейшую поддержку без вмешательства разработчиков. Веб-портал — сайт в компьютерной сети, который предоставляет пользователю различные приложения, которые работают в рамках этого сайта. Таким образом, создав библиотеку компонент(приложений), разработчик может решать разные бизнес-задачи, не затрачивая много времени на создание веб-сайтов, а собирая сайты из готовых компонент - портлетов.

Портлет — это отдельное небольшое веб-приложение, которое выполняется на портале, портал в свою очередь агрегирует один или несколько портлетов на отдельной веб-странице, которые обычно настраиваются для отдельных пользователей и групп портала. Итог - веб-страница, наполненная несколькими веб-приложениями. Простой пример портлета - корзина покупателя, создав лишь один раз такой портлет, появляется возможность использования данного портлета в других веб-сайтах, где необходим функционал корзины покупок.

Целью этой работы является рассмотрение принципов разработки и использования веб-порталов с многократно используемыми компонентами. Также, так как компоненты могут создаваться разными разработчиками, необходимо реализовать механизм комбинации нескольких портлетов в рамках одной веб-страницы.

I. ЯДРО ПОРТАЛА

Для функционирования веб-приложений и их агрегирования на одной странице необходим такой продукт как портал, в качестве портала был выбран готовый Open Source продукт - Liferay [1]. Liferay - программный продукт, представляющий собой корпоративный портал, то есть решение, предназначенное для централизованного доступа к нескольким различным корпоративным приложениям в одном месте. Liferay иногда описывается как система управле-

ния содержимым (CMS) или платформу для веб-приложений. Написан на языке Java и распространяется под двумя видами лицензий, свободной и проприетарной, используя бизнес-модель двойного лицензирования.

Liferay Portal позволяет пользователям настроить общий доступ к разным приложениям через один-единственный сайт. Это реализуется с помощью функциональных модулей, называемых портлетами. Портлеты должны иметь Backend и Frontend часть так как являются веб-приложениями.

Для автоматизации разработки портлетов необходимо простое решение, которое без вмешательства разработчика будет создавать из серверного и клиентского кода готовое приложение. В качестве языка для Backend разработки используется Java, а также фреймворк OSGi [2]. OSGi - состоит из двух частей. Первая часть представляет собой спецификацию модульных компонентов, называемых пакетами, которые обычно называются надстройками. Спецификация определяет инфраструктуру для жизненного цикла пакета и определяет, как пакеты будут взаимодействовать. Вторая часть OSGi - это реестр служб уровня виртуальной машины Java (JVM), который пакеты могут использовать для публикации, обнаружения и привязки к службам в сервис-ориентированной архитектуре (SOA) [3].

В результате получатся контроллеры, являющиеся промежуточной частью между клиентской частью и сторонним API или базой данных. Frontend часть реализуется на языке Angular - платформа веб-приложений с открытым исходным кодом на основе TypeScript. Результат компиляции Frontend кода - html файлы, отвечающие за отображение и js файлы, отвечающие за логику веб-приложения и обработку событий.

Был реализован фреймворк, который собирает результат компиляции Frontend и Backend кода, как единый портлет. Он основан на технологии JSP - технология, позволяющая веб-разработчикам создавать содержимое, которое имеет как статические, так и динамические ком-

поненты. Архитектура портлета представлена на рис.1.

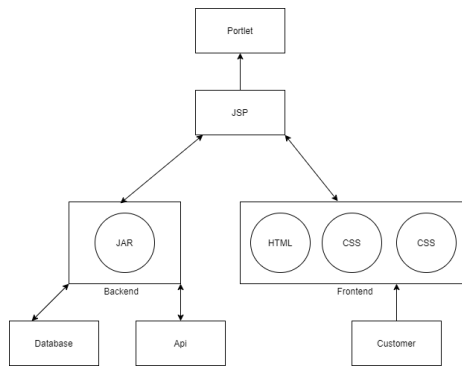


Рис. 1 – Архитектура разработки портлета

II. КОМБИНАЦИЯ ПОРТЛЕТОВ НА ВЕБ-СТРАНИЦЕ

Подразумевается, что разработчики будут реализовывать портлеты, добавлять их в общую библиотеку, после чего пользователи комбинируют портлеты из библиотеки составят свои веб-сайты. Однако, так как разные разработчики занимаются реализацией, у веб-приложений нету общего стиля и при добавлении разных портлетов, веб-страница не будет выглядеть как единое целое. Поэтому, был реализован функционал, позволяющий менять конфигурацию портлета в зависимости от содержимого веб-страницы.

При разработке портлета, необходимо выполнить конфигурацию портлета - заполнить XML файл, который регулирует цвет, шрифт, размер в зависимости от других компонент на странице. На рис.2 представлен пример веб-страницы, состоящей из трех компонент: график, статистика, описание. В левой части рис.2 видно как выглядела веб-страница до выполнения конфигурации портлетов, компоненты имеют разные цвета фона, размеры шрифта, веб-страница не выглядит как единое целое. После в файле конфигурации был настроен основной цвет веб-страницы (белый) и размеры шрифтов, компоненты изменили свой стиль, так как при их создании был реализован конфигурационный файл, и в правой части рис.2 можно видеть конечный

результат, который выглядит как единая веб-страница.

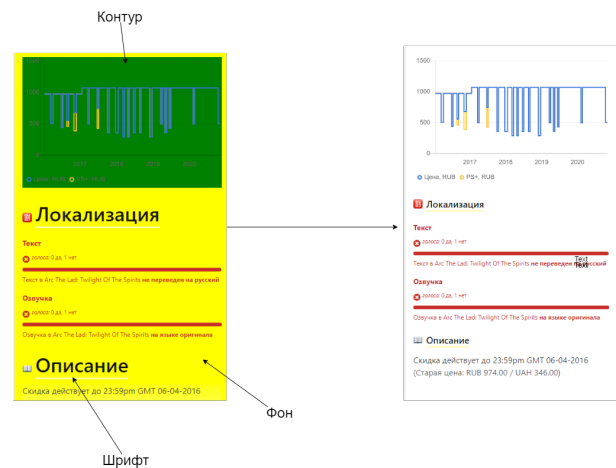


Рис. 2 – Пример конфигурации портлета

III. ЗАКЛЮЧЕНИЕ

Разработка веб-сайтов для решения бизнес-задач является финансово затратным процессом. Для упрощения реализации и сокращения затрат в данной работе предложен подход, который позволит пользователям создавать свои веб-сайты из разных веб-компонент при этом не требуя знания языков программирования.

В работе рассмотрена реализация веб-порталов с многократно используемыми компонентами. Описана технология создания портлетов и комбинации нескольких портлетов на веб-странице. Дальнейшее направление исследований включает возможность комбинации портлетов без вмешательства разработчика и пользователя, а также внедрения контент-менеджера для управления и версионирования контента веб-портала.

СПИСОК ЛИТЕРАТУРЫ

1. Liferay Portal Systems Development : Build Dynamic, Content-rich, and Social Systems on Top of Liferay / X. Y. Jonas . – Packt Publishing Limited, 2016. – 317 с.
2. Osgi in Action: Creating Modular Applications in Java / R. I. Hall . – Manning Publications, 2011. – 375 с.
3. OSGI [Электронный ресурс] / OSGI Alliance Community. – Режим доступа: <https://www.osgi.org/>. – Дата доступа: 12.05.2020.

АНАЛИЗ ПЛАТФОРМ ПОСТРОЕНИЯ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ИНТЕРФЕЙСОВ

Парамонова А. Е., Архипенко Я. С., Захарьев В. А.

Кафедра систем управления,

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: zahariev@bsuir.by

В условиях современного информационного общества люди ежедневно взаимодействуют с различными программными системами. В результате внедрения компьютерных систем во все сферы человеческой жизни все больше проявляется проблема перехода от визуальных и командных интерфейсов к естественно-языковым. В статье рассмотрены методы обработки естественного языка, структура цикла работы естественно-языкового пользовательского интерфейса, который включает такие стадии обработки, как морфологический, синтаксический и семантический анализ. Представлены основные платформы-конструкторы для построения виртуальных собеседников, их основные преимущества и сравнительный анализ.

ВВЕДЕНИЕ

Интеллектуализация прикладного программного обеспечения путем создания естественно-языковых интерфейсов является одной из главных тенденций развития информационных технологий в наше время. Это объясняется тем, что взаимодействие информационной системы и пользователя более эффективно, если пользователь формулирует запросы на естественном языке. Таким образом, подобные коммуникативные взаимодействия становятся возможными и без длительной специальной подготовки к ним.

СОДЕРЖАНИЕ ДОКУМЕНТА

Естественно-языковой интерфейс представляет собой достаточно сложную программу, состоящую из нескольких компонентов. В программе последовательно выполняется обработка и распознавание текста с помощью лингвистических методов анализа и специальных словарей. Цикл работы естественно-языкового пользовательского интерфейса начинается с ввода пользователем текста сообщения на естественном языке. Далее на основе введенного текста строится его формальное описание. Система хранит все результаты предыдущих анализов, что позволяет ей находиться в состоянии живого диалога с пользователем и избегать спорных ситуаций, связанных с использованием одних слов в разном контексте. В процессе обработки естественно-языкового текста последовательно выполняется морфологический, синтаксический и семантический анализ.

Виртуальный собеседник, чат-бот - программа, выясняющая потребности пользователей с целью их последующего удовлетворения, которая автоматически ведет общение с пользователем с помощью текста или голоса. Как любая интеллектуальная система, чат-бот имеет базу знаний. В простейшем случае она представля-

ет собой наборы возможных вопросов пользователя и соответствующих им ответов и реакций.

В сфере чат-ботов существует множество производителей, большинство из которых сосредоточены на программировании на естественном языке (NLP) и понимании естественного языка (NLU). В контексте искусственного интеллекта NLP охватывает несколько дисциплин, которые занимаются взаимодействием между компьютерными системами и естественным языком человека: дискурс-анализ, извлечение отношений, понимание естественного языка и другие области языкового анализа. Понимание естественного языка (NLU) - это подвид NLP, который фокусируется на понимании прочитанного и его семантическом анализе. Сегодня сочетание технологий NLP и NLU становится все более актуальным в различных областях программного обеспечения, включая технологии проектирования ботов.

Для персонализированного подхода к их построению можно воспользоваться онлайн-конструкторами, примерами могут служить такие платформы, как ManyChat, Leeloo.ai, Botsify, Dialogflow, Wit.ai/LUIS.

Dialogflow - это платформа для понимания естественного языка, используемая для разработки и интеграции диалогового пользовательского интерфейса в мобильные приложения, веб-приложения, устройства, боты, интерактивные системы голосового ответа и т. Dialogflow обеспечивает встроенную поддержку валют и даты, поддерживает большинство платформ, таких как Facebook Messenger, Slack, Alexa и Google Assistant. Wit.ai - это полностью бесплатная платформа, в том числе для коммерческого использования. LUIS (Language Understanding) - платформа Microsoft. Сервис на основе машинного обучения для встраивания естественного языка в приложения и ботов. Позволяет быстро создавать готовые к работе индивидуальные модели, которые имеют возможность постоянного совершенствования. SAP IRPA (Intelligent

Robotic Process Automation) – новый продукт компании SAP, созданный для автоматизации рутинных бизнес-процессов с помощью интеллектуальных ботов. Программа позволяет создавать, управлять и осуществлять мониторинг работы ботов. Для новых пользователей, являющихся сотрудниками партнеров компании SAP, доступна бесплатная версия приложения. IRPA-бот состоит из отдельных действий, которые составляют шаги процесса: открыть окно приложения, скопировать данные из ячейки. Далее окне визуализации процесса шаги соединяются между собой в нужном порядке и образуют сценарий, конечным результатом которого является выполнение бизнес-процесса. Выполнение созданного сценария может осуществляться с заданной нами периодичностью, вручную по команде пользователя или при выполнении определенных условий. Таким образом созданные боты при правильной настройке и интеграции в систему могут не только облегчить выполнение монотонных и трудоемких задач и повысить производительность, но и полностью автоматизировать многие задачи, выполняемые сотрудниками.

Группой экспертов оценивались самые популярные платформы для построения виртуальных собеседников по 4 показателям, значимость которых определяли пользователи, имевшие опыт в разработке чат-ботов: поддержка русского языка, интегрированный показатель возможностей NLP, возможность установки платформы on-premise, наличие графического редактора. Особое внимание было уделено возможностям искусственного интеллекта: обработке естественного языка, примерам использования каждой платформы, и отраслям, в которых платформа может найти применение. В таблице внизу приведены платформы, получившие наиболее высокие оценки по приведенным выше показателям, интервал оценивания [0,1]. Большая часть задач реализации и проектирования чат-ботов связана с формализацией и автома-

тической обработкой конструкций естественного языка и других коммуникационных процессов. В соответствии с этим актуальной является подготовка специалистов в области филологии, имеющих представление об особенностях организации интеллектуальных систем в общем и технологии создания естественно-языковых интерфейсов таких систем, в частности.

Таблица 1 – Сравнительный анализ платформ построения чат-ботов

Платформа	Русский языка	NLP	On-premise	Наличие графического редактора
IBM Watson	0.5	0.55	нет	есть
Google Dialog-flow	0.5	0.60	нет	есть
Facebook Messenger Platform	0	0.42	есть	есть
Amazon Lex	0	0.46	нет	есть
Baidu KITT.AI	0	0.38	нет	есть
Kore.ai	0	0.54	есть	есть
Bot-Engine.ai	0.35	0.38	нет	есть

СПИСОК ЛИТЕРАТУРЫ

1. Личаргин, Д. В. Методы и средства порождения семантических конструкций естественно-языковых интерфейсов программных систем / Д. В. Личаргин // ИВМ СО РАН – 2004. – с. 3–8.
2. Елисева, О. Е. Семантическое концептуальное проектирование естественно-языкового интерфейса интеллектуальной системы / О. Е. Елисева // БГУ-ИР – 2014. – с. 321–332.
3. Никонов, В. О. Диалоговая интеллектуальная система с естественно-языковым интерфейсом / В. О. Никонов // Кубанский ГТУ – 2007. – с. 8–22.

ПРИМЕНЕНИЕ ЭЛЕМЕНТОВ ВЫСШЕГО ПОРЯДКА В ФИЛЬТРАХ НИЖНИХ И ВЫСШИХ ЧАСТОТ

Петровский И. И., Свито И. Л.

Кафедра теоретических основ электротехники, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: Petrovskij@bsuir.by, svito@bsuir.by

В работе рассматривается возможность применения элементов высшего порядка, предложенных как новые элементы в теории электрических цепей, в различных устройствах автоматики и других электротехнических системах. Рассматривается возможность применения этих элементов в фильтрующих устройствах, в частности, в фильтрах высших и нижних частот.

ВВЕДЕНИЕ

Пассивные элементы электрических цепей R, L, C могут быть описаны уравнениями

$$F = (p^{(0;-1;1)}u, p^{(0;1;-1)}i) = 0 \quad (1)$$

при условии, что $p = d/dt$. Предложенные элементы высшего порядка, в которых применены многократно дифференцирующие и интегрирующие источники энергии, позволяют создать новые частотнозависимые, и с любой характеристикой элементы. Уравнение, описывающее эти элементы, может быть представлено в виде [1-3]:

$$F = (p^b u, p^a i) = 0 \quad (2)$$

I. ЭЛЕМЕНТЫ ВЫСШЕГО ПОРЯДКА В ФИЛЬТРАХ

Предположим, что эти элементы имеют линейную вольтамперную характеристику. Тогда можно это уравнение разрешить относительно напряжения или тока

$$u = K p^{a-b} i \quad (3)$$

$$i = 1/K p^{b-a} u \quad (4)$$

Так как a и b могут принимать любые целочисленные значения, то можно обозначить $a-b = n$. Тогда

$$Z(p) = \frac{U(p)}{I(p)} = K p^n \quad (5)$$

$$Y(p) = \frac{I(p)}{U(p)} = \frac{1}{K} p^{-n}. \quad (6)$$

Рассмотрим симметричные четырёхполюсники, например, фильтры Т- и П-образной структуры с элементами высшего порядка (рис. 1)

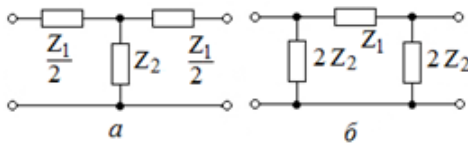


Рис. 1 – Фильтры Т- и П-образной структуры

Для Т- и П-образных схем примем сопротивления высшего порядка $Z_1 = K_1 p^{n_1}$ и $Z_2 = K_2 p^{n_2}$. При холостом ходе четырёхполюсника $A_{11} = ch(g) = \sqrt{\frac{K_1 p^{n_1}}{2K_2 p^{n_2}} + 1}$. Исследуем Г-образное звено фильтра $A_{11} = sh(\frac{g}{2}) = \pm \sqrt{\frac{K_1 p^{n_1}}{4K_2 p^{n_2}}}$ или $sh(\frac{g}{2}) = sh(\frac{a}{2} + j\frac{b}{2}) = sh(\frac{a}{2})\cos(\frac{b}{2}) + jch(\frac{a}{2})\sin(\frac{b}{2}) = \pm \sqrt{\frac{K_1 p^{n_1}}{4K_2 p^{n_2}}}$. Предположим, что правая часть уравнения при $p = j\omega$ чисто мнимая, т.е. $sh(\frac{a}{2})\cos(\frac{b}{2}) = 0$, то

$$ch(\frac{a}{2})\sin(\frac{b}{2}) = \pm \sqrt{\frac{K_1 p^{n_1}}{4K_2 p^{n_2}}}. \quad (7)$$

Условие выполняется в двух случаях, если $sh(\frac{a}{2}) = 0$ или $\cos(\frac{b}{2}) = 0$. Зона пропускания фильтра при $a = 0$, следовательно, $ch(\frac{a}{2}) = 1$, а фаза сигнала в зоне пропускания $\sin(\frac{b}{2}) = \pm \sqrt{\frac{K_1 p^{n_1}}{4K_2 p^{n_2}}}$. Так как и Т-образная и П-образная схемы фильтра могут быть представлены как две Г-образные, то фаза и затухание будут равны соответственно b и a .

Выражение $\pm \sqrt{\frac{K_1 p^{n_1}}{4K_2 p^{n_2}}} = \pm \sqrt{\frac{K_1}{4K_2}} p^{n_1-n_2} = pm \sqrt{\frac{K_1}{4K_2}} j^{n_1-n_2} \omega^{n_1-n_2}$, тогда мнимое, если $l = n_1 - n_2 = 4m - 2$, при этом $-\infty < m < \infty$.

Эти выводы можно пояснить с помощью систем координат, где на оси ординат откладываются элементы высшего порядка, которые находятся в продольной ветви Г-фильтра, на оси абсцисс откладываются элементы, лежащие в поперечной ветви.

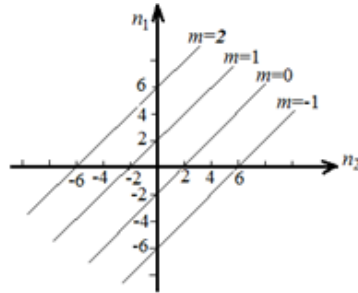


Рис. 2 – Элементы фильтра высших порядков

Для фильтра в области пропускания $a = 0$, выполняется условие $A_{11} = ch(a + jb) = ch(jb) = \cos(b)$, т.е. $-1 \leq A_{11} \leq 1$, а для фильтра с элементами высшего порядка $-1 \leq \frac{K_1 p^{n_1}}{4K_2 p^{n_2}} \leq 1$ или $-1 \leq \frac{K_1}{4K_2} j^{(n_1-n_2)} \omega^{(n_1-n_2)} \leq 1$.

Так как $j^{n_1-n_2} = -1$, то получаем $0 \leq \frac{K_1}{4K_2} \omega^{(n_1-n_2)} \leq 1$.

Исходя из этого условия, можно определить граничные частоты зоны пропускания. Если знак для $l = n_1 - n_2$ положительный, граничные частоты равны 0 и ω , что соответствует фильтру нижних частот. При l отрицательном граничные частоты равны ω_0 и l , что соответствует фильтру высших частот.

II. РЕАЛИЗАЦИЯ ФИЛЬТРА С ЭЛЕМЕНТАМИ ВЫСШЕГО ПОРЯДКА

Рассмотрим фильтр с элементами высшего порядка, представленный на рисунке 3.

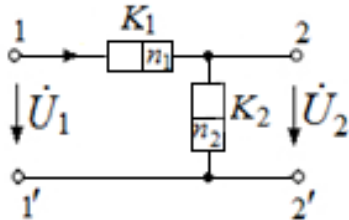


Рис. 3 – Фильтр с элементами высшего порядка

Передаточная функция:

$$\frac{U_2}{U_1} = \frac{\frac{K_1}{K_2}}{p^{n_1-n_2} + \frac{K_1}{K_2}}, \quad (8)$$

Если $n_1 - n_2 > 0$, то передаточная функция соответствует фильтру нижних частот, в другом случае при $n_1 - n_2 < 0$ после преобразования получим

$$\frac{U_2}{U_1} = \frac{p^{n_1-n_2}}{p^{n_1-n_2} + \frac{K_1}{K_2}}, \quad (9)$$

что соответствует фильтру высших частот.

В зависимости от разности между порядками элементов $l = n_1 - n_2$ коэффициенты затухания и коэффициенты фазы фильтров и высших частот представлены на графиках (рис. 4, 5).

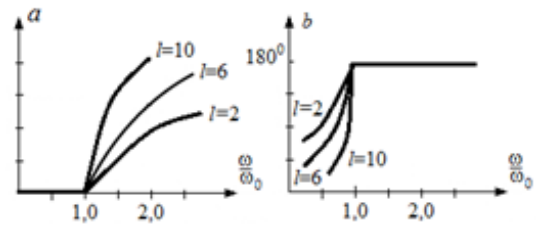


Рис. 4 – Характеристики ФНЧ

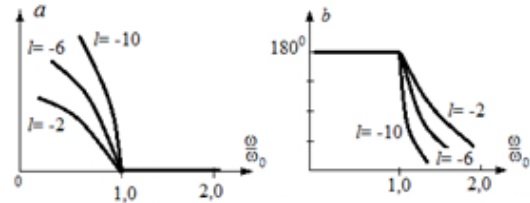


Рис. 5 – Характеристики ФВЧ

Соотношение между K_1 и K_2 определяет граничные (резонансные) частоты фильтра.

ЗАКЛЮЧЕНИЕ

Таким образом, из приведённого можно сделать вывод, что с помощью двух элементов высшего порядка можно получить любую передаточную функцию фильтров высших и нижних частот, а, следовательно, и получить желаемую характеристику коэффициента затухания и коэффициента фазы

СПИСОК ЛИТЕРАТУРЫ

1. Philippow E., Bruckner P., Schaltungsanordnung zum Erzeugung sowie zum Transformation linearer und nichtlinearer frequenzabhangiger Zweitpole hoherer Ordnung. Patentanmeldung, TH Ilmenau, 1976.
2. Атабеков Г. И. Теоретические основы электротехники. Линейные электрические цепи / Г. И. Атабеков // изд. ЛитРес –2009.
3. Батура М. П. Теория электрических цепей. / М. П. Батура, А. П. Кузнецов, А. П. Курулев // Минск. Высшая школа, –2007.
4. Шалфеев В. Д. Нелинейная динамика систем фазовой синхронизации. / В. Д. Шалфеев, В. В. Матросов // 2013. – 366 с.

СИНТЕЗ АЛГОРИТМОВ ИДЕНТИФИКАЦИИ СИСТЕМ С ФАЗОВЫМ УПРАВЛЕНИЕМ

Русак Л. В., Капанов Н. А.

Кафедра информационных систем и технологий, Международный институт дистанционного образования
Белорусский национальный технический университет

Кафедра систем управления, Белорусский государственный университет информатики и
радиоэлектроники

Минск, Республика Беларусь

E-mail: kapanov.nikolay@gmail.com

Рассматривается задача параметрической идентификации непрерывной линейной части системы с фазовым управлением на основе имеющейся априорной информации об уравнениях объекта в пространстве состояний и статистического анализа входных и выходных сигналов.

Введение

Среди множества актуальных задач синтеза систем с фазовым управлением (СФУ) следует особо выделить задачу идентификации. Под идентификацией в широком смысле понимается получение или уточнение по экспериментальным данным модели реального объекта (процесса), выраженной в тех или иных терминах (описанной на том или ином языке).

Среди различных задач идентификации и различных содержательных формулировок условий наблюдения рассмотрим задачу параметрической идентификации (оценки параметров) непрерывной линейной части (НЛЧ) СФУ на основе имеющейся априорной информации об уравнениях объекта в пространстве состояний и статистического анализа входных и выходных сигналов. Важность и актуальность выделения данной задачи обусловлены тем, что в настоящее время все большая часть исследований технических объектов проводится на основе использования их математических моделей [1].

Основной раздел

Пусть состояние НЛЧ СФУ описывается уравнением в форме Ланжевена со случайными параметрами

$$\dot{X}^{(s)}(t) = \varphi(X, D, s, t) + \sigma(X, D, s, t)U(t) + H(X, D, t)\xi(t) \quad (1)$$

при начальных условиях $X(t_0) = X_0$.

В данном случае $X^{(s)}(t)$ – вектор фазовых координат; $D = D^{(s)}(t)$ – n -мерный вектор в общем случае случайных параметров системы; $\sigma(X, D, s, t) = \sigma^{(s)}(X, t)$ – векторная функция; $H(X, D, s, t) = H^{(s)}(X, t)$ – матричная функция; $\xi(t)$ – вектор случайных возмущений; $U(t)$ – вектор детерминированных управлений. Уравнение измерителя следующее:

$$Z^{(l)}(t) = C^{(l)}(X, t) + a^{(l)}(t)\zeta(t) \quad (2)$$

Входящие в (1), (2) функции $\varphi(\dots)$, $C(\dots)$, а также характеристики шумов $\xi(t)$, $\zeta(t)$ считаются известными.

Области эксплуатации (работоспособности) системы и измерителя заданы в виде функциональных пространств:

$$X(t) \in X_{ex}, U(t) \in U_{ex}, Z(t) \in Z_{ex} \quad (3)$$

В результате проведения идентификационных экспериментов (натурных или моделированием) получены множества функций $X_u(t)$, $U_u(t)$, $Z_u(t)$, таких как

$$X_u(t) \in X_{exu} \subset X_{ex}, U_u(t) \in U_{exu} \subset U_{ex}, \\ Z_u(t) \in Z_{exu} \subset Z_{ex} \quad (4)$$

где X_{exu} , U_{exu} , Z_{exu} – пространства фазовых координат, управлений и измерений реальной системы при идентификационных экспериментах.

Задача идентификации состоит в том, чтобы на основе экспериментальных данных $X_u(t), U_u(t), Z_u(t)$ определить значение вектора параметров $\hat{D}^{(s)}(t)$, при котором разность $\hat{D}^{(s)}(t) = D^{(s)}(t) - \hat{D}^{(s)}(t)$ принимает наименьшее в определенной мере значение (обеспечить достаточную малость нормы разности). Для применения полученной в предыдущих подразделах теории необходимо случайный вектор $D^{(s)}(t)$ представить в соответствующем виде. Для широкого круга технических систем номинальные значения параметров $D_n(t)$ известны, а их фактические значения имеют малые отклонения от номиналов. Исходя из этого, на интервале наблюдения $[t_0, t]$ вектор $D^{(s)}(t)$ представим в виде

$$D^{(s)}(t) = D_n(t) + a_D^{(s)}(t) \quad (5)$$

где $a_D^{(s)}(t)$ – вектор малых отклонений параметров системы размерностью n_D .

В зависимости от модели реального объекта и условий исследований стохастическая модель вектора $a_D(t)$ может иметь различный вид: от $(\partial a_D / \partial t) = 0$ до $(\partial a_D / \partial t)$ – белый шум. Для широкого круга задач удобно применять модель

параметрических шумов, в которой компоненты вектора $a_D(t)$ описываются уравнением типа формирующего фильтра:

$$\dot{a}_D^{(s)}(t) = \frac{1}{\tau_D} a_D^{(s)}(t) + h_D \xi_D(t), a_D(t_0) = a_{D0} \quad (6)$$

где $\xi_D(t)$ – вектор белого шума с единичной матрицей интенсивностью E .

В данном случае τ_D – постоянная времени, характеризующая частоту флуктуаций параметров; матрица h_D характеризует величину отклонений параметров от номиналов.

Для идентифицируемой НЛЧ СФУ введем расширенный вектор состояний

$$X_p^{(s)T}(t) = \| X^{(s)}(t), a_D^{(s)}(t) \| \quad (7)$$

размерностью $n + n_D$, для которого уравнения состояния имеют вид

$$\begin{aligned} \dot{X}_p^{(s)}(t) = & \phi_p(X_p, D, s, t) + \sigma_p(X_p, D, s, t)U(t) + \\ & + H_p(X_p, D, s, t)\xi_p(t) \end{aligned} \quad (8)$$

при начальных условиях $X_p^{(s)}(t_0) = X_{p0}$.

$$Z_p^{(l)}(t) = C_p^{(l)}(X_p, t) + q^{(l)}(t)W(t). \quad (9)$$

В (8) входят блочные векторы

$$\begin{aligned} X_p^{(s)}(t) &= \begin{Bmatrix} X^{(s)}(t) \\ a_D^{(s)}(t) \end{Bmatrix}, \\ \xi_p(t) &= \begin{Bmatrix} \xi(t) \\ \xi_D(t) \end{Bmatrix}. \end{aligned}$$

При такой постановке задачи производится совместное оценивание и идентификация процесса (объекта) на основе использования в общем случае уравнений фильтрации вида:

$$-\Gamma_x(X, Z, t) + \Pi_x(X, Z, t) \quad (10)$$

$\Gamma_x(X, Z, t)$ и $\Pi_x(X, Z, t)$ характеризуют дискретную составляющую оценки процесса $X(t)$. Необходимо заменить $\hat{X}(t)$ на $\hat{X}_p^{(sl)}(t)$, а вектор оценки параметров в соответствии с определится следующим образом:

$$\hat{D}^{(sl)}(t) = D_n(t) + \hat{a}_D^{(sl)}(t). \quad (11)$$

Вектор оценки состояния и корреляционная матрица оценки точности в данном случае имеют вид

$$\hat{X}_p^{(sl)}(t) = \begin{Bmatrix} \hat{X}^{(sl)}(t) \\ \hat{a}_D^{(sl)}(t) \end{Bmatrix}, \hat{R}_p^{(sl)}(t) = \begin{Bmatrix} \hat{R}_{xx}^{(sl)}(t) & \hat{R}_{xa}^{(sl)}(t) \\ \hat{R}_{ax}^{(sl)}(t) & \hat{R}_{aa}^{(sl)}(t) \end{Bmatrix}.$$

В частном случае при непосредственном наблюдении вектора состояния и его точном измерении $X(t)$ может быть исключен из процесса оценивания (отнесен к детерминированному управлению). Следовательно, при неизменной структуре системы уравнения для оценки идентифицируемых параметров подсистемы в соответствии с используемым уравнением фильтрации имеют вид

$$\begin{aligned} \dot{\hat{X}}_p^{(sl)}(t) = & \int_{-\infty}^{+\infty} \phi_p(X_p, t) \hat{f}^T(X_p, t) dX_p - \\ & - \Gamma_x(X_p, Z, t) + \Pi_x(X_p, Z, t) - \\ & - \frac{1}{2} \int_{-\infty}^{+\infty} (X_p - \hat{X}_p) \rho(X_p, Z, t) \hat{f}(X_p, t) dX_p. \end{aligned} \quad (12)$$

$$\begin{aligned} \dot{\hat{R}}_p^{(sl)}(t) = & \int_{-\infty}^{+\infty} \text{div}([X(t) - \hat{X}(t)][X(t) - \\ & - \hat{X}(t)]^T \hat{\pi}(X_p, t)) dX + \int_{-\infty}^{+\infty} \hat{\pi}^T(X_p, t) \text{grad}([X(t) - \\ & - \hat{X}(t)][X(t) - \hat{X}(t)]^T) dX_p - \frac{1}{2} \int_{-\infty}^{+\infty} ([X(t) - \\ & - \hat{X}(t)][X(t) - \hat{X}(t)]^T - R) \rho(X_p, Z, t) dX_p - \\ & - \Gamma_R(X_p, Z, t) + \Pi(X_p, Z, t). \end{aligned} \quad (13)$$

Для вычисления функций $\Gamma_x(X_p, Z, t)$, $\Pi_x(X_p, Z, t)$, $\Gamma_R(X_p, Z, t)$ и $\Pi_R(X_p, Z, t)$ и практической реализации алгоритмов идентификации (12)–(13), как и при решении задач фильтрации, необходимо функцию $\hat{f}(X_p, t)$ аппроксимировать нормальным или усеченным нормальным распределением и использовать известные расчетные формулы [2]. При исследовании параметрической топологии (взаимного влияния параметров элементов систем с фазовым управлением) необходимо произвести идентификацию СФУ как сложной системы в целом, что для реальных систем в настоящее время может представлять неразрешимую задачу. Однако в перспективе актуальность данной задачи может возрасти по мере повышения возможностей измерителей и средств обработки информации.

1. Батура, М. П. Дискретные системы с фазовым управлением / М. П. Батура // Минск: Институт технической кибернетики НАН Беларуси, 2002. – 152 с.
2. Казаков, И. Е. . Методы оптимизации стохастических систем / И. Е. Казаков, Д. И. Гладков // Минск: Наука, 1987. – 304 с.

АНАЛИЗ ДИСКРЕТНЫХ СИСТЕМ МЕТОДОМ УРАВНЕНИЙ МОМЕНТОВ

Русак Л. В., Стасевич Н. А.

Кафедра информационных систем и технологий, Международный институт дистанционного образования
Белорусский национальный технический университет

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: stasevich@bsuir.by

Рассмотрены вопросы анализа дискретной системы методами уравнений моментов на примере системы первого порядка. При анализе данной дискретной системы будем следовать структуре исследования непрерывной системы фазовой автоподстройки частоты.

Введение

Среди множества актуальных задач синтеза систем с фазовым Составление и решение системы уравнений моментов для дискретных систем, в принципе, не отличается от аналогичных действий в случае непрерывных систем. Однако в каждом конкретном случае исследуемой системы процедуру составления системы уравнений моментов необходимо проводить заново, тогда как для непрерывных систем уже получено основное соотношение между моментами для системы, описываемой стохастическим дифференциальным уравнением произвольного порядка. Это и заставило ограничиться рассмотрением, в основном, непрерывных систем. Основные особенности метода уравнений моментов проявляются здесь наиболее выпукло, так как они могут быть рассмотрены независимо от процедуры составления системы уравнений моментов. Тем не менее, принципиальные вопросы анализа систем не зависят от того, дискретной или непрерывной она является. Это же относится и к качественным выводам о влиянии способов полиномиальной аппроксимации нелинейных характеристик на точность решения, получаемого методом уравнений моментов. Разумеется, при этом появляются специфические особенности. На наш взгляд, они важны и интересны, но не носят принципиального характера.

Основной раздел

Рассмотрим вопросы анализа дискретной системы методами уравнений моментов на примере системы первого порядка. При анализе данной дискретной системы будем следовать структуре исследования непрерывной системы фазовой автоподстройки частоты.

Пусть дискретная система описывается уравнением

$$x^+ = f(x) + \xi$$

где $x = x(t)$; $x^+ = f(x + 1)$; $\xi = \xi(t)$; $\xi \sim N(0, \sigma)$; $E\{\xi^2\} = \sigma$; $E\{x\xi\} = 0$; $t = 0, 1, 2, \dots$

Тогда основное соотношение между моментами примет вид:

$$\begin{aligned} E\{x^n\} &= E\left(\sum_{k=0}^n C_n^k \xi^k f^{n-k}(x)\right) = \\ &= \sum_{k=0}^n C_n^k E\{\xi^k\} E\{f^{n-k}(x)\} \end{aligned} \quad (1)$$

Основной интерес представляют системы уравнений возможно меньшего порядка. Не трудно показать, что в рассматриваемом случае все моменты нечетного порядка $E\{x^k\} = 0$. Поэтому можно ограничиться рассмотрением основных соотношений только между моментами четного порядка. Первые из упомянутых соотношений имеют вид:

$$\vartheta_2 = E\{f^2(x)\} + \sigma;$$

$$\vartheta_4 = E\{f^4(x)\} + 6\sigma E\{f^2(x)\} + 3\sigma^2; \quad (2)$$

$$\begin{aligned} \vartheta_6 &= E\{f^6(x)\} + 15\sigma E\{f^4(x)\} + \\ &+ 45\sigma^2 E\{f^2(x)\} + 15\sigma^3; \end{aligned} \quad (3)$$

$$\begin{aligned} \vartheta_8 &= E\{f^8(x)\} + 28\sigma E\{f^6(x)\} + \\ &+ 210\sigma^2 E\{f^4(x)\} + 420\sigma^3 E\{f^2(x)\} + 105\sigma^4. \end{aligned} \quad (4)$$

Здесь использовано выражение высших моментов центрированного нормального распределения величин ξ через второй центральный момент (2). При этом речь идет не о нормализации (решения), а используется свойство распределения помехи, которое по определению нормальное.

Для того чтобы рассматриваемая дискретная система имела непосредственное отношение к рассмотренной ранее системе ФАПЧ первого порядка, исследуем случай, когда

$$f(x) = x - F(x) \cong 1/3!x^3 - 1/5!x^5.$$

Как уже отмечалось, для получения системы уравнений моментов из основных соотношений между моментами необходимо представить все нелинейные функции, входящие в упомянутые

соотношения, конечными отрезками степенных рядов. Если иметь в виду систематическое использование рядов Тейлора при исследовании системы, то для получения полиномиальной аппроксимации любой функции в произвольной степени достаточно располагать конечным отрезком ряда Тейлора только этой функции. Как и в непрерывном случае, ограничимся тремя членами ряда Тейлора при разложении дискриминационной характеристики (4) ($k < 4$) и примерно таким же числом уравнений моментов.

При ограничении двумя членами ряда Тейлора ($k = 2$) в разложении дискриминационной характеристики $f(x) = 1/3!x^3$, $E\{f^i(x)\} = 1/6^i\theta_{3i}$. Тогда уравнения (2) – (4) принимают вид:

$$\theta_2 = 1/6^2\theta_6 + a; \theta_4 = 1/6^4\theta_{12} + 1/6a\theta_6 + 3a^2; \quad (5)$$

$$\theta_6 = 1/6^6\theta_{18} + 15/6^4a\theta_{12} + 45/36a^2\theta_6 + 15a^3. \quad (6)$$

Систематическое использование рядов Тейлора предполагает отбрасывание высших моментов. Это означает, что из одного уравнения моментов может быть определен только второй момент θ_2 , из двух уравнений – два момента θ_2 и θ_4 из трех – три и т.д. В условиях настоящего примера определение двух моментов θ_2 и θ_4 по двум уравнениям (5) не приносит ничего нового в полученный ранее результат (при $k = 1$, $n = 1$), так как в полном соответствии с изложенной выше методикой следует положить $\theta_6 = \theta_{12} = 0$. Поэтому для получения следующего приближения ($k = 2$) рассмотрим три первых уравнения моментов. При $\theta_{12} = \theta_{18} = 0$ они принимают вид

$$\begin{aligned} \theta_2 &= 1/6^2\theta_6 + a; \theta_4 = 1/6a\theta_6 + 3a^2; \\ \theta_6 &= 45/36a^2\theta_6 + 15a^3. \end{aligned} \quad (7)$$

Отсюда получим

$$\theta_2 = a + \frac{5a^3}{12 - 15a^2}. \quad (8)$$

При учете трех членов ряда Тейлора ($k = 3$) $f(x) = 1/6x^3(1 - 1/20x^2)$; $E\{f^i(x)\} = 1/6^i E\{x^{3i}(1 - 1/20x^2)^i\}$.

Отсюда следует:

$$\begin{aligned} E\{f^2(x)\} &= 1/6^2 E\{x^6(1 - 1/20x^2)^2\} = \\ &= 1/6^2(\theta_6 - 1/10\theta_8 + 1/400\theta_{10}) \end{aligned} \quad (9)$$

$$\begin{aligned} E\{f^4(x)\} &= 1/6^4 E\{x^{12}(1 - 1/20x^2)^4\} = \\ &= 1/6^4(\theta_{12} - 1/5\theta_{14} + 6/400\theta_{16} - \dots) \end{aligned} \quad (10)$$

$$E\{f^6(x)\} = 1/6^6(\theta_{12} - \dots);$$

$$E\{f^8(x)\} = 1/6^8(\theta_{24} - \dots)$$

Очевидно, что моменты $E\{f^6(x)\}$ и $E\{f^8(x)\}$ являются линейными комбинациями моментов не ниже 18-го и 24-го порядков соответственно. Если при анализе системы ограничиться тремя уравнениями моментов, то всеми

моментами выше шестого порядка придется пренебречь.

В этом случае система уравнений (2) – (4) с учетом выражений (9) – (10) примет вид системы уравнений (7). Другими словами, решение получившейся системы уравнений моментов будет иметь вид выражения (8). Это означает, что третий член разложения дискриминационной характеристики не может быть учтен при использовании только трех уравнений моментов.

Применение метода уравнений моментов для анализа дискретных систем, в принципе, почти не отличается от применения этого метода для анализа непрерывных систем. При этом, правда, использовались только ряды Тейлора для полиномиальной аппроксимации всех нелинейных зависимостей, входящих в основные соотношения между моментами.

Покажем, что сказанное относится и к методу нормализации. В этом случае моменты высших порядков исследуемой координаты не отбрасываются, а выражаются через низшие моменты. В нем рассмотрена последовательность примеров, некоторые из которых можно почти дословно повторить, чтобы проиллюстрировать применимость этого метода не только для непрерывных, но для дискретных систем.

Пусть дискриминационная характеристика представлена двумя членами ряда Тейлора, т.е. $k = 2$ Тогда уравнение (2) принимает вид уравнения (5). Вместо отбрасывания момента θ_6 , как это делалось раньше, можно выразить этот момент через момент второго порядка α некоторого распределения. В соответствии с выражением (2) положим $\theta_6 = 5!!\alpha^3$. Тогда уравнение примет вид $\theta_2 = 1/6^2\theta_6 + a = 15/36\alpha^3 + a$.

Аналогично, при $k = 3$ получим

$$\theta_2 = 1/6^2(\theta_6 - 1/10\theta_8) + a = 1/36(15\alpha^3 - 10,5\alpha^4) + a. \quad (11)$$

Сравним последние выражения с аналогичными выражениями для непрерывных систем. Продолжим аналогию и в части их использования для анализа системы. Положив $\theta_2 = \alpha$, получим, что второй центральный момент (дисперсия θ_2) может быть найден из решения алгебраического уравнения четвертой степени, которое следует из (11) при данном значении параметра α .

Различия в применении метода уравнений моментов к непрерывным и дискретным системам имеются только на этапе составления основных соотношений между моментами. Для дискретных систем они имеют более громоздкий вид, из-за чего без соответствующего программного обеспечения вычисления могут быть доведены до конца только в случае систем невысокого порядка [1].

1. Батура, М. П. Дискретные системы с фазовым управлением / М. П. Батура // Минск: Институт технической кибернетики НАН Беларуси, 2002. – 152 с.

МЕТОДЫ ПЛАНИРОВАНИЯ ТРАЕКТОРИИ В НЕИЗВЕСТНОЙ СРЕДЕ

Снисаренко С. В.

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: snisarenko@bsuir.by

Представлен обзор и приведена сравнительная характеристика локальных методов планирования траектории и поиска пути в динамически изменяющейся рабочей среде для задачи управления коллаборативным роботом.

ВВЕДЕНИЕ

Одной из ключевых задач автоматического управления коллаборативным роботом является задача планирования оптимальной траектории. Робот в реальном режиме времени, в реальных условиях рабочей среды должен самостоятельно решать задачу планирования пути без вмешательства человека. Рабочая среда, в которой находится робот, обычно содержит динамические препятствия (человек, наличие других роботов в рабочей области, оборудование и т.д.), движение в ней по предварительно заданной траектории практически невозможно. В случае динамически изменяющейся рабочей среды необходимо онлайн - планирование траектории движения на основе поступающей информации в навигационную систему о динамической модели рабочего пространства.

Актуальный алгоритм планирования траектории, в частности – обхода препятствия, должен сочетать в себе преимущества глобального и локального методов планирования пути. Глобальный метод позволяет планировать глобальный путь на основе глобальной информации об окружающей среде, а локальные методы позволят роботу избегать столкновений с препятствиями, которые не были обнаружены заранее или находятся в движении, так что их текущее положение не может быть установлено заранее. Проанализируем наиболее распространенные и представляющие интерес локальные методы для построения оптимальной траектории в онлайн – режиме для робота.

I. МЕТОДЫ ПОТЕНЦИАЛЬНЫХ ПОЛЕЙ

Методы потенциальных полей основаны на физической аналогии с движением заряженной частицы в электростатическом поле. Препятствия сцены генерируют отталкивающие силы, а целевая точка маршрута – значительную притягивающую силу. Направление и скорость движения тела определяются градиентом потенциального поля. Навигационная функция задается суммой притягивающих и отталкивающих потенциалов: $U(C) = U_{att}(C) + U_{rep}(C)$. При продвижении робота к целевой точке C_{goal} функ-

ция притягивающего потенциала U_{att} должна монотонно возрастать, однако ее конкретный вид может существенно варьироваться. Основным недостатком методов потенциальных полей, использующих градиентный спуск, является проблема локальных минимумов. Эта проблема в частности решена в известном методе рандомизированных потенциальных полей (Randomized Potential Fields). [1]

II. МЕТОД BUG3

Из семейства методов Bug представляет интерес следующий алгоритм, учитывающий функцию дистанционного определения препятствий, строящий локальные графы касательных. Алгоритм использует следующие понятия: точка разрыва (препятствие/граница видимости) и интервал непрерывности (определяется двумя точками разрыва). Агент движется в сторону цели по прямой до тех пор, пока не зафиксирует препятствие на пути (прямая будет пересекать интервал непрерывности). После этого агент начинает движение к одной из точек разрыва, для которой эвристическая оценка минимальна (например, к той, сумма расстояний от которой до робота и до цели минимальна). В процессе движения агент получает новые точки разрыва и движется к ним до тех пор, пока эвристическая оценка не перестанет уменьшаться (т.е. не достигнет локального минимума). После этого агент двигается вдоль границы препятствия, сохраняя направление. Покидает границу, когда препятствие больше не мешает проходу к цели. Данный подход позволяет найти более оптимальный путь, чем предыдущие, однако требует более сложной организации системы технического зрения. [2]

III. МЕТОДЫ ГИСТОГРАММЫ ВЕКТОРНОГО ПОЛЯ (VFH)

Метод гистограммы векторного поля (vector field histogram, VFH) лишен недостатков предыдущих методов. В настоящее время он является одним из наиболее популярных для локального планирования, и используется в режиме управления реальным временем. Три основных этапа:

1) составление сетки на двумерной декартовой плоскости, описывающей препятствия во круг робота, перестроение сформированной ги стограммы на одномерную полярную.

2) Выбор наиболее подходящего сектора с низкой плотностью препятствий, вычисление угла поворота в этом направлении.

Существуют некоторые улучшенные модификации метода VFH: VFH+ и VFH*. Метод VFH+ учитывает размер робота, динамические ограничения и соответствующее расширение размеров препятствий, увеличивает гладкость спланированной траектории. Метод VFH* считается улучшением VFH и VFH+, учитывает глобальную информацию об окружении, выбирает наилучшее направление движения с совместным использованием глобального алгоритма A*. VFH* подходит для работы с неточной информацией, использует вероятностный подход и считается быстрым и надежным в среде с большим количеством препятствий, однако существуют неразрешимые ситуации при наличии U-образных и симметричных препятствий. [3]

IV. МЕТОДЫ НА ОСНОВЕ СПЛАЙН - ИНТЕРПОЛЯЦИИ

Рассчитываемая в ходе выполнения алгоритмов траектория должна быть гладкой. Для описания степени гладкости кривой используют символ C^m означающий, что функции, представляющие кривую, имеют непрерывные производные до n порядка включительно. Построенный путь должен удовлетворять ограничениям робота. Робот не может изменять скорость или направление движения мгновенно. Также следует учитывать ограничения, вызываемые предельными углами поворота звеньев и скоростью их вращения. Для удовлетворения этих требований траектория движения должна быть гладкой кривой класса C^2 . Если же добавить еще ограничений, например, поставить условие непрерывности графика кривизны траектории, то это будет равносильно критерию гладкости кривой класса C^3 . При сглаживании пути используют следующие подходы: 1. Корректировка опорных точек.

2. Сглаживание ломаных в сочленениях. Траектория манипулятора строится из множества кривых. Ломаные участки траектории корректируются только в окрестностях стыка кривых.

Методы сглаживания ломаных участков траектории:

а) метод кривых Дубинса (англ. – Dubins). Суть метода заключается в замене частей отрезков, прилегающих к углу ломаной, дугой окружности;

б) метод полярных многочленов, сглаживающая траектория, задаваемая многочленом от угла в полярных координатах, используется для

решения проблемы разрывом функции кривизны;

в) кривые Безье, кубические кривые Безье рассчитываются с использованием четырех опорных точек, которые выбираются на звеньях ломаной, кривизна полученной сглаженной траектории будет иметь разрыв. В отличие от использующих кубические кривые, траектории, сглаженные с помощью дважды дифференцируемой кривой Безье (класса C^2), не имеют разрывы в функции кривизны. Для расчета такой кривой используются семь опорных точек;

г) кубический B-сплайн, использование кубического B-сплайна позволяет добиться непрерывного, гладкого изменения кривизны. Для этого необходимо использование не менее пяти опорных точек. Однако число контрольных точек может быть увеличено, что позволит увеличить степени свободы в выборе формы кривой.[4]

ЗАКЛЮЧЕНИЕ

Метод потенциальных полей позволяет построить гладкий путь. Реализация метода относительно проста и ясна. Преимуществом метода является малое время расчета. Поэтому его часто используют при локальном планировании, когда расчеты должны выполняться в режиме реального времени. Но при использовании этого метода робот может попасть в точку локального минимума и, в конечном счете, не достичь цели. Оптимизационные методы имеют преимущество в том, что с помощью них можно учесть самые разные ограничения, например динамические или ограничения на управление. Но с увеличением размерности пространства состояний и области окружающей среды, с увеличением степени детализации модель быстро усложняется, что приводит в результате к замедлению процесса вычисления. Реализовать оптимизационные алгоритмы в режиме реального времени не всегда представляется возможным. Целесообразно использовать гибридные, комбинированные методы глобального и локального онлайн - планирования траектории робота в зависимости от конструкции робота, выполняемых задач и состояния окружающей среды.

СПИСОК ЛИТЕРАТУРЫ

1. Казаков К.А., Семенов В.А. Обзор современных методов планирования движения. Труды ИСП РАН, 2016 г., стр. 241-294. DOI: 10.15514/ISPRAS2016-28(4)-14
2. Бекасов Д.Е. Применение аппарата нечеткой логики при решении задачи поиска пути в неизвестном окружении. Молодежный научно-технический вестник №05, май 2012, 77-48211/446707
3. Ulrich, I. VFH: Local obstacle avoidance with lookahead verification /I. Ulrich, J. Borenstein. // Proceedings of the IEEE International Conference on 55 Robotics and Automation. – San Francisco, CA, USA: – 2000. – Vol. 3. – С. 2505- 2511.
4. Роджерс, Д. Математические основы машинной графики. / Д. Роджерс, Д. Адамс. – Мир, 2001.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ IM VIEW ДЛЯ ИССЛЕДОВАНИЯ ТЕПЛОВЕНТИЛЯЦИОННЫХ РЕЖИМОВ РАБОТЫ АСИНХРОННЫХ ЭЛЕКТРОДВИГАТЕЛЕЙ

Третьяков А. С.

Кафедра «Электропривод и автоматизация промышленных установок», Межгосударственное образовательное учреждение высшего образования «Белорусско - Российский университет»
Могилев, Республика Беларусь
E-mail: loggie121@gmail.com

В статье обоснована важность учета тепловентиляционных режимов работы асинхронных электродвигателей при их эксплуатации. Рассмотрен вопрос своевременного профилактического обслуживания и ремонта электрических машин для их долгой и безотказной работы. Предложена концепция и описание программного обеспечения IM View для оценки, контроля и прогнозирования тепловых режимов асинхронных электродвигателей.

ВВЕДЕНИЕ

Одним из важных факторов длительной и безотказной работы асинхронного электродвигателя является обеспечение его номинального теплового режима работы. Эксплуатируя асинхронные электродвигатели в установленных заводом-изготовителем условиях, а также проводя плановые профилактические осмотры, можно получить длительный срок безотказной работы электрических машин. Система ГОСТов полностью определяет виды испытаний, сроки их проведения, а также порядок операций при проведении испытаний.

I. КРАТКОЕ ОПИСАНИЕ ТЕПЛОВЫХ ИСПЫТАНИЙ

Одним из испытаний является измерение температур отдельных узлов асинхронных электродвигателей, и испытание асинхронных электродвигателей на нагревание. Тепловые испытания электрических машин проводятся в режимах, наиболее приближенных к их номинальным (исходя из условий испытаний). При этом испытание можно начинать как с практически холодного, так и с нагретого состояния машины. Для сокращения продолжительности испытания машину допускается перегрузить в начале испытания, насколько это допустимо из соображений ее механической и электрической прочности.

При испытании асинхронного электродвигателя на нагревание надлежит измерять:

1. Напряжения и токи статора, их частоту, подводимую электрическую мощность, и мощность на валу;
2. Частоту вращения, или скольжение асинхронных двигателей;
3. Температуру отдельных узлов машины;
4. Прочие величины, которые могут оказывать влияние на нагревание испытуемой машины.

Проведение тепловых испытаний проводится разными способами, например:

1. Испытание непосредственной нагрузки;
2. Испытание на нагревание методом двух частот;
3. Испытание методом эквивалентной нагрузки, и тд.

Выбор способа определяется условиями и целями тепловых испытаний. Конечным этапом является получение экспериментальных температур согласно метода испытаний.

II. ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ IM VIEW

Одной из программ, предназначенных для экспериментального определения, или расчета температур отдельных узлов асинхронного электродвигателя, является программное обеспечение IM View. Скриншот программы представлен на рис.1. Написанная на кроссплатформенном фреймворке qt5, программа может быть портирована на платформы Windows и Mac Os.

На главном окне программы располагается пять вкладок:

1. Вкладка «Исходные данные»;
2. Вкладка «Идентификация параметров»;
3. Вкладка «Энергетические показатели»;
4. Вкладка «Тепловентиляция»;
5. Вкладка «Выходные данные».

Вкладка «Исходные данные» представляет собой по сути базу данных, где хранятся все исходные данные по каждому испытанному двигателю. В настройках программы предусмотрена возможность ограничения прав пользователей базы данных.

Вкладка «Идентификация параметров» представляет собой подпрограмму для идентификации параметров T-образной схемы замещения асинхронного электродвигателя. Весь математический аппарат основан на методике, предложенной украинским ученым Д. Л. Приступой, доработан и модернизирован [1].

Вкладка «Энергетические показатели» представляет собой подпрограмму для экспе-

риментального определения параметров энергетической диаграммы испытуемого асинхронного электродвигателя, визуального отображения электрических и механических величин с их последующим сохранением в массивы [2].

Вкладка «Тепловентиляция» представляет собой подпрограмму, которая либо снимает показания с датчиков температур, либо на основании измеренных данных идентифицирует температуры отдельных узлов асинхронного электродвигателя [2].

Вкладка «Выходные данные» представляет собой набор настроек для создания итогового отчета и сохранения требуемых данных в массивы.

III. ОСОБЕННОСТИ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ IM VIEW

В основе программы IM View лежит алгоритм тепловых испытаний, описанный в п.2, а также математический аппарат, представленный в [2]. Для ее корректной работы на данном этапе реализуется менеджер сеансов. Это сделано для того, чтобы в рамках одного сеанса все настройки, экспериментальные и расчетные данные были задокументированы, упакованы и сохранены в одном месте с приложенным итоговым отчетом по данному сеансу.

Весь математический аппарат (дифференциальные уравнения), представленный в данной программе, переведен на систему разностных уравнений. Это дает выигрыш времени расчета и обработке данных, а также сокращении объема аппаратных ресурсов, необходимых для решения задач программы.

Реализована возможность экспериментального, или ручного расчета температур конкретно взятого асинхронного электродвигателя.

Под данную программу разработан цифровой блок ввода налоговых сигналов, а также отдельное окно для его настройки и калибровки, что в связке дает единый программно - аппаратный комплекс для исследования тепловентиляционных режимов асинхронных электродвигателей.

IV. ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ IM VIEW

Одним из основных направлений развития данного программного обеспечения является дальнейшая оптимизация программного кода с целью его интеграции в прошивку системы управления преобразователем частоты для идентификации, контроля и прогнозирования температур отдельных узлов асинхронного электродвигателя.

Вторым направлением может быть прогнозирование остаточного теплового ресурса изоляции обмоток статора электрических машин.

V. СПИСОК ЛИТЕРАТУРЫ

1. Пересада, С. М. Алгоритм идентификации электрических параметров асинхронного двигателя на основе адаптивного наблюдателя полного порядка: синтез и экспериментальное тестирование / С. М. Пересада, С. Н. Ковбаса, Д. Л. Приступа // Праці Інституту електродинаміки Національної академії наук України. – 2013. – № 34. – С. 27–34.
2. Третьяков, А. С. Моделирование тепловентиляционных режимов работы асинхронных электродвигателей при питании от синусоидального источника напряжения / А. С. Третьяков, О. А. Капитонов // Вестн. ГГТУ им. П. О. Сухого. – 2018. – №2 (73). – С. 66–73.

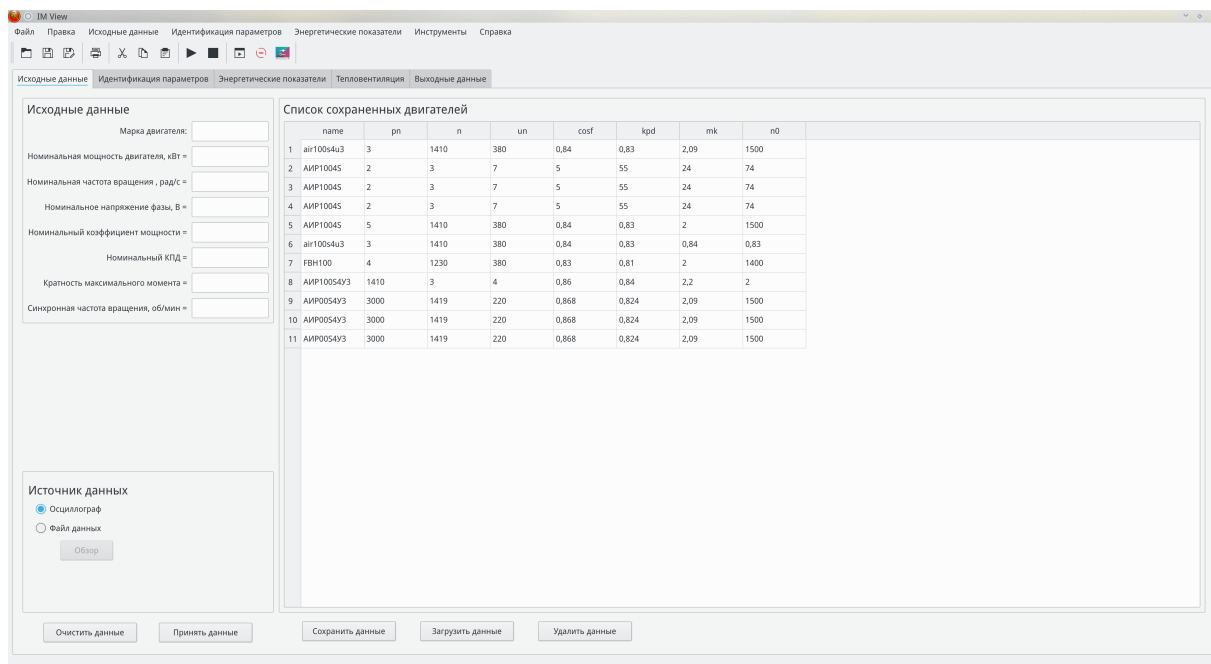


Рис. 1 – Скриншот программного обеспечения IM View

СЕРВОПРИВОД С НЕЛИНЕЙНЫМ КОМБИНИРОВАННЫМ УПРАВЛЕНИЕМ

Хаджинов М. К., Доманов А. Т., Павлова А. В.

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: pavlova@bsuir.by

Рассмотрен сервопривод подчинённого регулирования с тремя входами для комбинированного управления координатой, скоростью и ускорением. Осуществлен синтез задатчика комбинированного управления, в который введено упреждение в сигналы управления для частичной компенсации инерционности сервопривода. Приведены варианты настроек задатчика и результаты моделирования работы сервопривода с задатчиком

Введение

Современные автоматические системы высокой динамической точности обычно строят на основе принципа комбинированного управления, сочетающего в себе принципы управления по отклонению и по возмущению. Точность работы комбинированных систем выше точности систем, использующих только один из принципов управления, причем недостатки обоих принципов при их объединении устраняются.

Высокоточный сервопривод обычно имеет структуру подчинённого регулирования с отдельными регуляторами тока, скорости и координаты. Очевидно, что координированное управление сразу тремя сигналами на входах всех регуляторов эффективнее управления только одной координатой. Представляет несомненный интерес введение нелинейных функций в каналы комбинированного управления. В первую очередь это касается канала комбинированного управления по скорости. Помимо этого для реализации комбинированного управления скоростью, ускорением и координатой необходимо создать устройство или алгоритм для вычисления скорости и ускорения из управляющего сигнала координатой – задатчик комбинированного управления. Нелинейные функции комбинированного управления могут быть введены как в задатчик управления, так и в саму систему управления сервоприводом.

I. Постановка задачи

Ставится задача разработки нелинейного алгоритма комбинированного управления, пригодного для практического применения. Естественно, на наш взгляд, выглядит решение задачи с использованием в алгоритме нелинейной квадратичной функции с сохранением знака. Такое решение целесообразно использовать для канала комбинированного управления скоростью. Более того, такое решение естественно вписывается в систему управления сервопривода в контур оптимизации по быстродействию [1]. Для канала комбинированного управления ускорением

предлагается использовать как кубическую зависимость, так и квадратичную с сохранением знака. Динамические характеристики сервопривода с использованием нелинейной квадратичной функции с сохранением знака можно улучшить, если задатчик будет вырабатывать выходные управляющие сигналы ускорения, скорости и координаты с упреждением входного сигнала управления координатой [2].

II. Структура задатчика

Структура задатчика имеет вид следящего оценивающего фильтра с модальным регулятором. Простейший следящий фильтр второго порядка можно составить из последовательно включённых двух интеграторов для оценивания скорости и координаты и модального регулятора, обеспечивающего отслеживание управляющего сигнала координаты и желаемую динамику оценивания. Выходы интеграторов дадут фильтрованные оценки скорости и координаты, а входной сигнал первого интегратора даст нефильТРованную оценку ускорения. Таким образом будет сформировано три сигнала для управления тремя регуляторами сервопривода. Улучшить динамику сервопривода можно соответствующей настройкой модального регулятора. Для этого в структуру следящего фильтра добавляем инерционное звено после второго интегратора, тогда ошибка следящего фильтра будет формироваться как разница между сигналом управляющей координаты и задержанным инерционным звеном сигналом оценки координаты самого фильтра. Порядок фильтра увеличивается до трёх. Модальный регулятор для задатчика с упреждением можно настроить на динамику, близкую к динамике задатчика второго порядка без упреждения.

III. Результаты моделирования

За базу для настройки модального регулятора задатчика выбрана ω_{tk} – частота среза контура тока сервопривода. Три регулятора сервопривода настраиваются, как единый модальный

регулятор на полином Латропа и обеспечивают переходную характеристику с перерегулированием 1.55%. Частота ω_0 масштабирования полиномов настройки модальных регуляторов задатчиков выбиралась больше ω_{tk} в три раза. Частота ω_u упреждения (излома ЛАХ апериодического звена в задатчике) была выбрана меньше ω_{tk} в 2 раза.

Комбинированное управление сервоприводом от задатчика управления эффективно работает в режиме слежения. В переходных режимах ему свойственно большое перерегулирование. Дополнительным фактором существенно влияющим на процессы управления является физическое ограничение уровня управляющих сигналов на входе объекта управления. Это ограничение в первую очередь проявляется на переходных процессах устранения рассогласования начальных условий при вхождении в режим слежения. Эффективность сервопривода с нелинейным комбинированным управлением проверялась моделированием в Симулинке. Сравнивались процессы управления сервопривода с нелинейным и линейным комбинированным управлением для двух зон: в линейной зоне ограничений и нелинейной. Схема моделирования сервопривода с нелинейным комбинированным управлением приведена на рис.1.

По результатам выборочного моделирования системы управления с несколькими нелинейностями можно отметить следующее:

Замена линейного управления в канале комбинированного управления по скорости на квадратичное с сохранением знака для процессов в линейной зоне ограничителя управления ускоряет процессы примерно в полтора раза.

Но одинаковая длительность процесса вхождения в режим слежения у квадратичного комбинированного управления достигалась при в 5 раз более жёстком ограничении на управление по сравнению с линейным комбинированным управлением.

Сигналы комбинированного управления по ускорению отсекаются нелинейностью ограничения на входе контура тока, и комбинированное управление эффективно лишь в режиме слежения. При этом линейное оно или нелинейное не имеет существенного значения в практических приложениях.

ЗАКЛЮЧЕНИЕ

Замена линейного управления в канале комбинированного управления по скорости на квадратичное с сохранением знака эффективна как для управления в линейной зоне ограничения сигналов управления, так и на этапе устранения рассогласования начальных условий.

Линейное или нелинейное комбинированное управление по ускорению не имеет существенного значения в практических приложениях.

СПИСОК ЛИТЕРАТУРЫ

1. Хаджинов, М.К. Сервоприводы с квадратичной обратной связью по скорости / М.К. Хаджинов, А.Т.Доманов, А.В.Павлова // Информационные технологии и системы 2017 (ИТС 2017): материалы международной научной конференции, БГУИР, Минск, Беларусь, 25 октября 2017 - С. 72-73. <https://libeldoc.bsuir.by/handle/123456789/20570/>
2. Хаджинов, М.К. Сервопривод с комбинированным управлением / М.К. Хаджинов, А.Т.Доманов, А.В.Павлова // Информационные технологии и системы 2019 (ИТС 2019): материалы международной научной конференции, БГУИР, Минск, Беларусь, 25 октября 2019 - С.96 -97. <https://libeldoc.bsuir.by/handle/123456789/37538/>

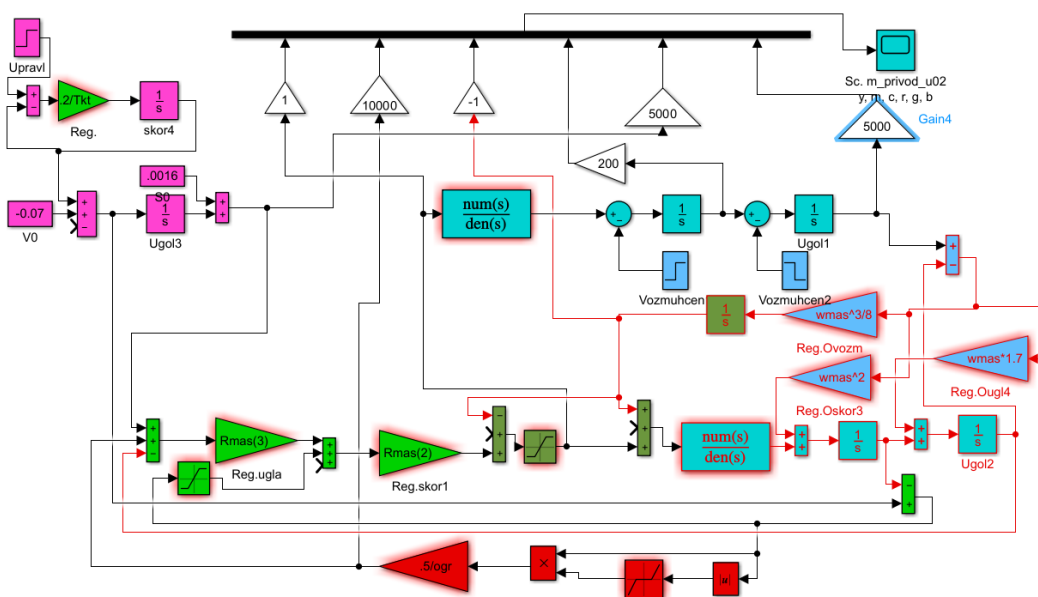


Рис. 1 – Схема моделирования сервопривода с нелинейным комбинированным управлением

ПАРАМЕТРИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИМПУЛЬСНЫХ СИСТЕМ ФАЗОВОЙ СИНХРОНИЗАЦИИ

Шилин Л. Ю., Кукин Д. П.

Факультет информационных технологий и управления, Белорусский государственный университет
информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: kukin@bsuir.by

В докладе рассмотрены вопросы параметрического проектирования импульсных систем фазовой синхронизации, предложен обобщенный метод синтеза указанных систем на основе построения областей захвата и удержания, затронуты вопросы построения универсальной математической модели рассмотренного класса устройств.

ВВЕДЕНИЕ

В настоящее время широкое распространение получили импульсные системы фазовой синхронизации (ИСФС). Это системы высокочастотной и тактовой синхронизации, формирующие опорные колебания на приемной стороне непосредственно по принимаемому сигналу, а также синтезаторы частот, осуществляющие формирование сетки высокостабильных колебаний задающих генераторов. Подобные системы применяются в радиопередающей и радиоприемной аппаратуре при демодуляции сигналов, в системах синхронизации при передаче дискретных сообщений, в доплеровских измерительных системах [1, 2].

ПРИНЦИПЫ ПАРАМЕТРИЧЕСКОГО ПРОЕКТИРОВАНИЯ ИСФС

Одной из важнейших задач проектирования ИСФС является определение полос захвата и удержания. В настоящее время существуют аналитические, машинно-ориентированные и экспериментальные методы определения полос захвата и удержания [1]. Авторами предложена эффективная методика расчета ПЗ систем произвольного порядка, базирующаяся на универсальной цифровой модели ИСФС. Точность и эффективность предложенного метода определяются характеристиками математической модели рас-

чета переходных процессов в ИСФС. В основе предложенного метода лежат известные методы математического моделирования ИСФС [3, 4, 5], которые дополнены и модифицированы для создания универсальной цифровой модели анализа полос захвата и удержания ИСФС. В ходе анализа разнообразных структурных схем ИСФС авторами были выделены наиболее общие принципы построения этого класса устройств [6, 7] и получена обобщенная структурная схема, которая приведена на рисунке 1. На рисунке 1 введены следующие обозначения: ИФД – импульсно-фазовый детектор; НЛЧ – непрерывная линейная часть; ОУ – объект управления; ОС – цепь обратной связи; $y(t)$ – входной сигнал; $u(t)$ – сигнал обратной связи; $\epsilon(t)$ – сигнал рассогласования; $e(t)$ – управляющий сигнал; $z(t)$ – выходной сигнал ОУ; $g(t)$ – постоянная составляющая ОУ; $\omega(t)$ – выходной сигнал системы. В представленной выше структурной схеме в качестве ИФД могут быть использованы разнообразные импульсные (в том числе и цифровые) устройства с широким спектром дискриминационных характеристик [8]. НЛЧ включает звенья коррекции и фильтрации [9]. ОУ представляет собой существенно нелинейный элемент, как правило, это синхронизатор, двигатель, управляемый напряжением генератор либо фазовозвращатель [10].

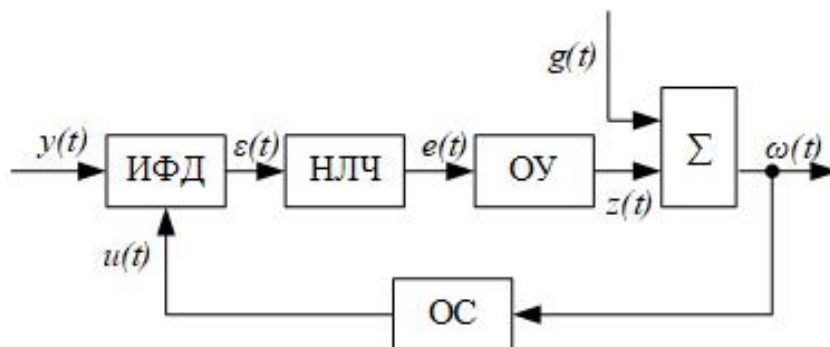


Рис. 1 – Пример конвейера обработки данных с последовательными блоками

Авторами получено следующее уравнение замкнутой системы:

$$\int_{nT+\tau_n}^{nT+kT+\tau_n+k} \omega(t)dt = 2\pi j N_D, \quad (1)$$

где N_D – коэффициент деления цепи ОС; k – количество импульсов входного сигнала за время τ_n ; j – количество импульсов на выходе ОС за время $t \in [nT + \tau_n; nT + kT + \tau_n]$. Величина j определяется из соотношения:

$$j = \text{int}[\phi_n/2\pi] + 1, \quad (2)$$

где ϕ_n – набег фазы сигнала $u(t)$ на интервале времени $t \in [nT + \tau_n; nT + kT]$:

$$\phi_n = \frac{1}{N_D} \int_{nT+\tau_n}^{nT+kT} \omega(t)dt, \quad (3)$$

Величина k определяется из соотношения:

$$k = \text{int}[\tau_n/T] + 1. \quad (4)$$

Выходной сигнал системы $\omega(t)$:

$$\omega(t) = z(t) + g(t) = M(e(t)) + g(t), \quad (5)$$

где $M(e(t))$ – модуляционная характеристика объекта управления. Установившийся режим работы системы характеризуется постоянством координаты x_{n+1} , неизменностью длительности импульсов τ_{n+1} , набега фазы ϕ_{n+1} и др.:

$$z_{n+1} = z_n = z^*;$$

$$x_{n+1} = x_n = x^*;$$

$$\tau_{n+1} = \tau_n = \tau^*;$$

$$\phi_{n+1} = \phi_n = \phi^*;$$

Такое математическое описание ИСФС позволяет осуществить программное моделирование переходных процессов во всех сложных режимах работы систем, что открывает возможность проведения расчета полос захвата и удержания. Важнейшей проблемой является задача определения сходимости переходного процесса. Авторами предложена методика программного анализа сходимости динамического процесса, учитывающая особенности процессов в ИСФС. С помощью предложенной методики авторами получены зависимости полос захвата и удержания от значенных параметров системы.

ЗАКЛЮЧЕНИЕ

Таким образом, авторами предложена методика, позволяющая производить параметрическое проектирование ИСФС основываясь на анализе полос захвата полученной путем программного моделирования. В основе метода лежит разработанный принцип математического описания ИСФС, характеризующийся применением совокупности как имитационного, так и аналитического подхода.

СПИСОК ЛИТЕРАТУРЫ

1. Шалфеев В. Д. Нелинейная динамика систем фазовой синхронизации. / В. Д. Шалфеев, В. В. Матросов // 2013. – 366 с.
2. Леонов Г. А. Системы фазовой синхронизации в аналоговой и цифровой схемотехнике. / Г. А. Леонов, С. М. Селеджи // Невский Диалект 2003. – 112 с.
3. Батура М. П. Принципы построения программных систем фазовой синхронизации. / М. П. Батура и др. // Информационные технологии и системы 2014 (ИТС 2014) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 29 октября 2014 г. – Information Technologies and Systems 2014 (ITS 2014) : Proceeding of The International Conference, BSUIR, Minsk, 29th October 2014 / редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2014. – С. 20-21.
4. Батура М. П. Принципы математического описания систем фазовой синхронизации. // Международная научно-техническая конференция, приуроченная к 50-летию МРТИ-БГУИР (Минск, 18-19 марта 2014 года) : материалы конф. В 2 ч. Ч. 1. – Минск, 2014. – С. 448-449.
5. Кукин Д. П. Моделирование цифровых устройств фазовой синхронизации / Доклады БГУИР – 2010. – №5 (76).
6. Кукин Д. П. Синтез цифровых устройств фазовой синхронизации с учетом требований по устойчивости быстрдействию и точности / Информационные технологии и системы 2011: материалы международной научной конференции. БГУИР, Минск, 2011.
7. Кукин Д. П. Автоматизированное проектирование цифровых устройств фазовой синхронизации / Материалы международной конференции ИТС 2013, БГУИР МИНСК, 2013 // Д. П. Кукин, Д. Л. Шилин.
8. Принципы построения математических моделей систем фазовой синхронизации / М. П. Батура [и др.] // Доклады БГУИР. – 2014. – № 2 (80). – С. 177-185.
9. Батура М. П. Анализ сложных режимов работы устройств фазовой синхронизации / М. П. Батура и другие // Информационные технологии и системы 2013 (ИТС 2013) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 23 октября 2013 г. – Information Technologies and Systems 2013 (ITS 2013) : Proceeding of The International Conference, BSUIR, Minsk, 24th October 2013 / редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – С. 20–21.
10. Шилин Д. Л. Статистическая динамика импульсных систем фазовой синхронизации / Д. Л. Шилин, Л. Ю. Шилин // Доклады БГУИР. – 2009. – № 8 (46). – С. 40-44.

ОБЗОР СХЕМОТЕХНИЧЕСКИХ РЕАЛИЗАЦИЙ СИСТЕМ УПРАВЛЕНИЯ АККУМУЛЯТОРНЫХ БАТАРЕЙ

Яскевич Д. Н., Довнар А. Д., Курулёв А. П.

Кафедра теоретических основ электротехники, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: andrevdovnar@gmail.com

В работе приведён краткий обзор основных функциональных блоков систем управления литий-ионных аккумуляторных батарей. Предложена классификация этих систем по схемотехнической реализации функциональных блоков и алгоритм выбора подходящей реализации для практического применения.

Введение

Аккумуляторные батареи (АКБ) применяются в широком спектре устройств от носимой электроники до электротранспорта. На сегодняшний день более широкое применение получили литий-ионные АКБ, т.к. они обладают наилучшими показателями по отношению запасаемой энергии к массе (плотности энергии), количеству циклов перезаряда и отсутствию эффекта памяти в сравнении с другими типами АКБ. Однако высокая плотность энергии в литий-ионных АКБ приводит к необходимости использовать их в определённых диапазонах токов заряда/разряда, температур и напряжений, задаваемых производителем. Несоблюдение этих режимов работы приводит к ускоренной деградации аккумуляторов, их выходу из строя, а порой и к воспламенению.

I. Системы управления АКБ

Для контроля за напряжениями ячеек, их балансировкой, током и температурой батареи используют системы управления аккумуляторных батарей (далее в тексте называемые BMS, от английского Battery management system) [1]. Существуют различные схемотехнические реализации BMS. Однако все они имеют схожую структуру, обобщённую на рис.1.

Структурная схема состоит из:

1. блока мониторинга и балансировки – измеряет напряжения параллельных групп ячеек, отвечает за их балансировку;
2. блока измерения тока – измеряет ток заряда/разряда батареи;
3. логического исполнительного блока – получает информацию о напряжении ячеек, токе и температуре батареи и в случае несоответствия параметров заданному режиму работы, подаёт команду об отключении на блок коммутации;
4. блока коммутации – по команде отключает батарею от нагрузки или зарядного устройства, исключая возможность эксплуатации батареи вне заданного режима работы.

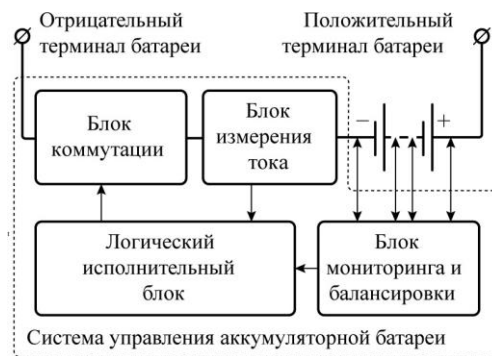


Рис. 1 – Обобщённая структурная схема BMS

По схемотехнической реализации BMS можно разделить на аналоговые (в документации на ИС приводится название Battery Secondary Protector) и цифровые. Цифровые в свою очередь можно поделить на централизованные (обычно называемые Battery Monitor или Fuel Gauge) и модульные, состоящие из ведущего модуля (Master) и нескольких ведомых измерительных модулей (Slave) [2].

II. Аналоговая BMS

Структурная схема аналоговой BMS приведена на рис. 2. Блок коммутации реализуют на МОП-транзисторах. Ток измеряется по падению напряжения на токоизмерительном резисторе.

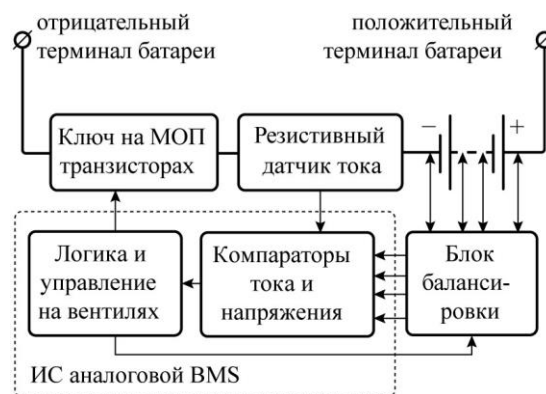


Рис. 2 – Структурная схема аналоговой BMS

Напряжения ячеек и токоизмерительного резистора поступают на входные компараторы интегральной схемы (ИС), настроенные на нужный диапазон безопасных значений – они выполняют функцию каскада мониторинга. При выходе из этого диапазона компаратор сигнализирует об ошибке в логический блок, реализованный на вентилях, который подаёт команду на закрывание ключей или в случае расхождения напряжений соседних ячеек, фиксируемого другими компараторами, включает балансировку. Такой относительно простой элементной базой обусловлена низкая цена аналоговых BMS, однако эта реализация не позволяет обмениваться данными с другими системами, что ограничивает функционал только контролем параметров и балансировкой. Примеры ИС, используемых в аналоговых BMS: HY2113, BQ2970, R5640G, BQ29209-Q1.

III. ЦИФРОВАЯ ЦЕНТРАЛИЗОВАННАЯ BMS

Структурная схема цифровой централизованной BMS представлена на рис. 3. В отличие от аналоговой, где напряжения ячеек и ток сравнивались с внутренним опорным напряжением компаратором, цифровая реализация подразумевает аналого-цифровое преобразование этих параметров, а затем сравнение их численных значений с записанными в памяти диапазонами. Это позволяет программно настраивать рабочий диапазон токов и напряжений и использовать одну BMS для разных батарей. Измерения проводятся специализированными ИС мониторинга, например, STC3100, MAX17853, LTC2941, BQ27010. Результаты измерений поступают в микроконтроллер, который обрабатывает их, управляет ключами и балансировкой, отвечает за индикацию и связь, являясь ядром цифровой системы управления. Хранение данных в цифровом виде позволяет реализовать функции быстрой зарядки, расчёта процента заряда, индикации других параметров батареи и передачи этих данных другим системам. Однако, такие BMS как правило значительно дороже аналоговых из-за использования более сложных цифровых схем и микроконтроллера.

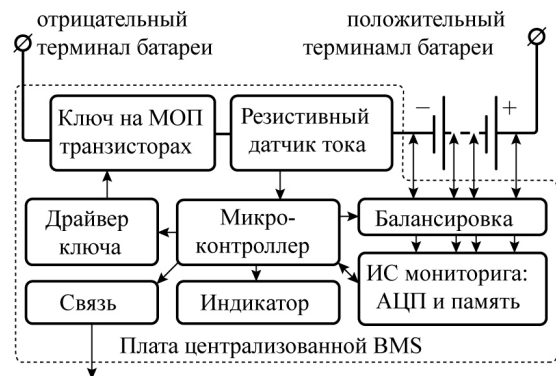


Рис. 3 – Структурная схема централизованной цифровой BMS

IV. ЦИФРОВАЯ МОДУЛЬНАЯ BMS

Как правило модульные цифровые BMS (рис.4) используются в самых больших батареях, например, в электротранспорте. Такие батареи зачастую разбивают на модули, соединённые последовательно. Тогда каждому модулю соответствует своя Slave BMS, полностью выполняющая роль блока мониторинга и балансировки (рис.1). Slave модули строятся на базе ИС мониторинга, схожих с приведёнными выше, однако не содержат МК. Примеры ИС bq76PL455A, BQ79606A-Q1, LTC6813-1. Батареи электротранспорта рассчитаны на большие токи, поэтому для коммутации используют контакторы (управляемые реле), а для измерения тока используют датчики Холла. Логический и исполнительный каскад представлен платой Master BMS, построенной на базе микроконтроллера, который получает информацию о параметрах батареи, обрабатывает её, отображает пользователю и управляет ключами. Платы модульных BMS достаточно сложны и обычно разрабатываются производителями отдельно для конкретных применений и имеют наибольшую стоимость из всех перечисленных топологий.

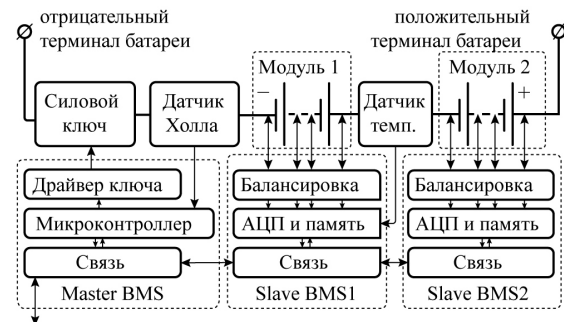


Рис. 4 – Структурная схема модульной цифровой BMS

Для выбора схемотехнической реализации BMS предлагается следующий алгоритм:

- определить конфигурацию батареи – количество ячеек, соединённых последовательно и параллельно;
- решить, будет ли использоваться модульная конструкция АКБ;
 - да: применяется цифровая модульная реализация BMS;
 - нет: решить, необходимы ли функции расчёта и индикации параметров:
 - * да: применяется цифровая централизованная реализация BMS;
 - * нет: применяется аналоговая реализация BMS.

1. M. Lelie, T. Braun, Battery Management System Hardware Concepts – 2018 MDPI, Basel, Switzerland.
2. Зайцев А.С., Смирнов А.А., Бутарович Д.О. – Обзор топологий современных систем управления зарядом накопителей энергии для электрических транспортных средств. Журнал автомобильных инженеров №2(109) 2018. ISSN 20739133

Секция «Интеллектуальные
информационные технологии»

Section «Intellectual information
technologies»

DEEP FEATURES CREATION FOR SMILE CLASSIFICATION IN BIOMETRIC SYSTEMS

Boris Assanovich, Yury Veretilo
YK State University of Grodno
National Anti-Doping Laboratory
Grodno, Minsk, Republic of Belarus
E-mail: bas@grsu.by, y.veretilo@antidoping.by

The new face deep features obtained from video with use of autoencoder have been proposed and a smile imprint for the biometric cryptosystem was constructed. The performance of the proposed system has been verified with the use of posed smiles dataset from UvA-NEMO database.

INTRODUCTION

Recently, the analysis of human emotions has attracted more and more researchers. Often a person's smile is a key factor not only in determining the person's psychological state, but also has become one of the most preferred biometric methods both in video surveillance and in digital banking [1].

To ensure high accuracy of recognition and resistance to various attacks (cropping, spoofing, etc.), a number of authors switched to the analysis of the dynamic characteristics of a person's face or to soft biometrics. Recently Murat Taskiran et al [2] have applied dynamic face features extraction from videos and used them for face recognition. Authors performed the segmentation of smile phases from video and extraction of the main face landmarks allowed to recognize face with the equal error rate $EER = 30\%$ (threshold 2.8) for the spontaneous smile and with $EER = 21.25\%$ (threshold 2.39) for the posed one with the use of 648 feature vector. Different techniques use facial dynamics to identify a person with the use of various spatio-temporal parameters of face. Classical linear methods of image processing and feature extraction based on principal components analysis (PCA) are replaced by non-linear transformations. Compared with PCA, the use of an autoencoder significantly increases the classification accuracy.

In this paper, we will consider the use of auto-encoders to extract soft biometric data from a person's smile and use them for authentication and identification to provide access to digital services.

The article provides the analysis of classification efficiency of face deep features and demonstrates the creation of a digital smiling face imprint to provide a secure biometric interface.

I. AUTOENCODERS

Actually an autoencoder is the network that is forced to learn to select the most important features from data [3]. To exclude the process of overfitting in ML sparse autoencoders are used.

In our system we applied so-called stacked autoencoder (SAE) that is a neural network

including several layers of sparse autoencoders where output of each hidden layer is connected to the input of the successive hidden layer. In this case the hidden layers are trained in an unsupervised way and then fine-tuned by a supervised method.

The use of SAE will allow the use of the trained data of the output layer of the neural network as features for biometric separation of users into genuine and imposters (persons who pretends to be somebody else).

The more the two distributions are separated and the smaller the standard deviation for each distribution, the better the separation of the classified classes. This property of distributions is estimated by such a parameter as decidability index

$$DI = \frac{|\mu_g - \mu_i|}{\sqrt{(\sigma_g^2 + \sigma_i^2)/2}}, \quad (1)$$

where μ_g, μ_i and σ_g, σ_i are the means and standard deviations of genuine and imposter distributions.

In addition to the decidability index, an equal error rate (ERR), which is the error rate at which a false accept rate (FAR) is equal to a false rejection rate (FRR), is normally used as a measure of biometric system verification accuracy. In biometrics FAR is the rate at which an imposter print is incorrectly accepted as genuine and FRR is the rate at which a genuine print is incorrectly rejected as imposter.

Then, a comparison of users by their feature representation can be performed by the evaluation of a receiver operator characteristic (ROC) curve that demonstrates how a genuine acceptance rate (GAR) determining a correctly received genuine fingerprint depends on a change in FAR.

Biometric authentication is the process of establishing user identity by measurements of his biological characteristics. To handle the variability inherent in biometric authentication, it is necessary to create and store a template for each user. To create it, one can use the fuzzy commitment scheme with application of error correcting codes (ECC) [4].

II. PROPOSED SYSTEM

As soon as autoencoders are widely used in many applications many researches have tested their efficiency. The experiments of face recognition based on SVM and softmax classifiers have shown that in recognition of 51 classes autoencoder achieved the error rate 9.53% while PCA 13.5% [5].

Inspired by a possible improvement in the classification characteristics, we used an auto-encoder to obtain biometric data on a person's smile and bind them to a secure user key. According to our knowledge, the use of auto-encoders to create a biometric imprint from face dynamics like a human smile has not been considered before.

Thus, the user smile imprint captured from video can serve as the biometric key to organize the access to different external digital services.

III. RESULTS AND CONCLUSIONS

In order to perform the features extraction and processing from captured images and video we developed the FaceAnalyzer Application (FAA) that has been applied to select and encode face HOG features, extract and normalize frames related to three main phases of video smile, calculate the statistical characteristic of biometric features.

The UvA-NEMO Smile Database has been used for experiments. It contained videos of spontaneous and deliberate smiles collected from 400 subjects.

Using FAA, video was analyzed with a person's smile and its three phases:

1. onset (neutral to expressive) phase;
2. apex;
3. offset phase (expressive to neutral), describing the state of a smiling person's face.

To compare the opportunities of binary and non-binary ECC by analogy with our earlier work [4], we extracted 4464-dimensional vectors from selected video smile frames, calculated their reliability and applied a 511-element mask for subsequent encoding with BCH codes (511,58,91) (511,28,111). The results showed a superposition of intraclass and intraclass distributions on each other and significant classification errors.

A series of experiments were performed with SAE to get good compact biometric features. To reduce time spent, in these experiments the subsets of 40 subjects randomly selected from the entire UvA-NEMO Database were used, reproducing a posed smile. Then normalized grayscale images from corresponding video of 112x112 pixels in size, scaled to 50%, creating a vector length of the input layer of 6272 elements have been used for unsupervised learning of SAE.

The training of the SAE layers consisted of 2 stages:

1. Unsupervised training on a $6 \times 40 = 240$ frames from training dataset, containing 3 phases of a smile and a neutral state;

2. Supervised training, when 4 additional video frames were added to SAE input from the control dataset for tuning testing.

Soft-Max classifier was used for test identification.

To assess the quality of training, on the basis of latent layer data y , such values as FRR, FAR, GAR, ERR and DI were calculated and the ROC-characteristic was monitored, as well as the values of MSE for controlling the intraclass and interclass distribution. The results of SAE training and tuning through the corresponding number of iterations are shown in Fig.1, and the histograms of the obtained distributions in Fig.2.

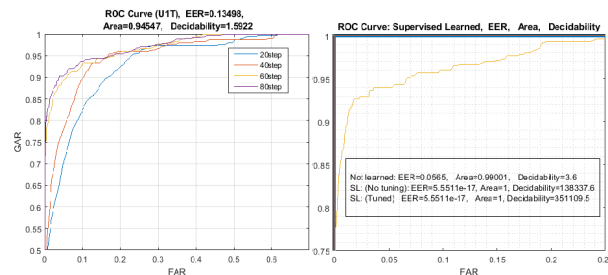


Fig. 1 – ROCs

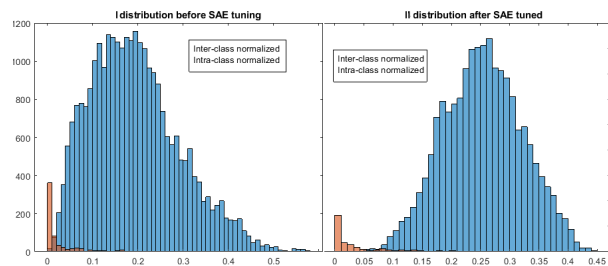


Fig. 2 – DIs

The experiments based on of UvaNemo and Caltech datasets performed have shown the improvement in the accuracy of classification genuine smile from posed one by 10%, as well as a reduction in the complexity of biometric templates design.

1. Cook, S. Selfie banking: is it a reality? Biometric Technology Today, 3(March). 2017 P. 9–11.
2. Taskiran, M. et al. Face Recognition Using Dynamic Features Extracted from Smile Videos. In: IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, 3-5 July 2019. P. 1-6.
3. Assanovich B. AutoEncoders for Denoising and Classification Applications. OSTIS 2020, M., 2020. – Вып. 4. – P. 309-312
4. Assanovich, B., Veretilo Yu. Biometric Database Based on HOG Structures and BCH Codes. In Proc. Information Techn. and Syst. BSUIR, 2017 P. 286-287.
5. Siwek, K., Osowski, S. Autoencoder versus PCA in face recognition. In: 18th International Conference on Computational Problems of Electrical Engineering (CPEE). Kutna Hora, 11-13 September 2017 . P. 1–4.
6. Assanovich B., Bich N., Pronevich. A. Smile Biometric Imprint Creation with the Use of Autoencoder. TC3И. – M., 2020 . – P. 8.

АДАПТИВНАЯ СИСТЕМА ОБУЧЕНИЯ АНГЛИЙСКОМУ ЯЗЫКУ НА ОСНОВЕ КЛАСТЕРНОГО АНАЛИЗА

Войтиховский П. Д., Сафоненко К. А., Давыденко И. Т.

Факультет информационных технологий и управления,

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: voitich97@gmail.com

В статье рассмотрены существующие подходы к разработке систем обучения английскому языку, а также отличия традиционных и адаптивных систем обучения. Приведены достоинства и недостатки каждого из подходов.

ВВЕДЕНИЕ

В современном мире знание иностранных языков является необходимостью, которая обуславливается различными факторами. В первую очередь, английский язык является стандартом международной коммуникации, что является причиной использования этого языка на конференциях, в профессиональной коммуникации, в написании профессиональной литературы и научно-исследовательских работ. Второй причиной для изучения иностранных языков является то, что появляется возможность чтения художественной и технической литературы в оригинале. Это особенно актуально для людей, работающих с быстро развивающимися технологиями, когда ожидание перевода документации стоит очень дорого.

Процесс изучения иностранных языков является довольно длительным, так как он затрагивает такие аспекты, как изучение лексики, грамматики, произношения, а также тренировку восприятия иностранной речи и способов самовыражения на иностранном языке. В то же время у людей зачастую нет необходимого количества времени для того, чтобы качественно проработать каждый из этих аспектов. В результате этого появляются различные проблемы, такие как языковой барьер, когда человеку тяжело выразить свои мысли, отсутствие понимания различных диалектов языка или непонимание культурных особенностей, которые ведут за собой к непониманию различных фраз. В связи с этим, критически важной задачей является сокращение времени, затрачиваемого на каждый из аспектов изучения языка.

Существует множество различных причин, которые влияют на время изучения языка. К таким причинам можно отнести то, что пользователям необходимо самостоятельно искать всю необходимую для изучения информацию. Время, затрачиваемое на поиск информации увеличивается с каждым днем, так как ее количество ежедневно растет, а автоматизированных способов выделять релевантную информацию нет, поэтому эта задача возлагается на плечи обучающихся. Следовательно, актуальной является задача

разработки средств обучения иностранному языку, которые бы позволили сократить время подбора материалов и ускорить процесс обучения при самостоятельном обучении.

I. СУЩЕСТВУЮЩИЕ ПОДХОДЫ К ИЗУЧЕНИЮ ИНОСТРАННОГО ЯЗЫКА

На сегодняшний день существуют различные способы изучения иностранного языка, многие из которых реализованы в виде различных программных обучающих систем.

Данные системы можно разбить на следующие категории:

Обучение с учителем. Данный тип обучения подразумевает обучение с преподавателем. Он может быть реализован в виде индивидуальных или групповых занятий. Этот тип обучения был одним из самых первых способов изучения иностранного языка. К групповым занятиям можно отнести занятия в школах, университетах, к индивидуальным - занятия с репетитором.

Самостоятельные занятия. Данный тип систем направлен на то, чтобы у каждого пользователя была возможность на организацию самостоятельного изучения языка. Примерами таких систем могут быть:

1. Busuu [1] - это не одно, а целый набор приложений, которые позволяют отработать различные навыки. Данные приложения позволяют тренировать лексику путем изучения слов и добавления их в словарь, грамматику путем изучения правил и выполнения упражнений, письмо, аудирование и другое;
2. Lingualeo [2] - приложение, использующее игровую форму обучения. Оно позволяет изучать слова и фразы, составлять собственный словарь, добавлять к словам озвучивание, тренировать грамматику и общаться с другими пользователями системы. Выполнение заданий дает очки, которые необходимы для дальнейшего изучения;
3. Duolingo [3] - популярное мобильное приложение для изучения различных иностранных языков с нуля. Для изучения представ-

ляются шесть языков: английский, немецкий, португальский, французский, итальянский и испанский. Программы всех курсов построены в форме деревьев достижения. Чтобы пройти на следующий уровень, нужно набрать определенное количество очков на предыдущем уровне. Очки даются за правильные ответы. Приложение позволяет делиться своими результатами с другими людьми и сравнивать их с собой;

4. Rosetta Course [4] - данное приложение организует изучение иностранного языка путем построения ассоциаций. Это приложение является идеальным вариантом для людей, которые не хотят тратить время и силы на механическое заучивание грамматических и лексических структур.

Online обучение с преподавателем.

Данный вид изучения языка подразумевает использование современных информационных технологий для того, чтобы связаться с человеком, который будет обучать иностранному языку.

Данные системы не реализуют ничего нового с точки зрения подачи материала и способов их усвоения и отработки. Они предоставляют те же упражнения на лексику и грамматику, выделяют такое же количество времени на отработку навыков произношения и понимания иностранного языка, а занятия также проводятся либо в группах, либо индивидуально.

Новизна данных систем заключается в том, что они обеспечивают возможность работать с преподавателями, находящимися в другой части мира, или самостоятельно, имея определенные средства для обучения. Это открывает безграничные возможности по подбору преподавателей. Так, можно выбрать для себя преподавателя, который будет лучше подходить обучаемому по темпераменту, методам и способам обучения. Более того, можно выбрать в качестве преподавателя носителя языка, что поможет погрузиться в языковую среду и быстрее начать понимать носителей языка, которые не говорят на понятном для обучаемого акценте.

II. АДАПТИВНЫЕ СИСТЕМЫ ОБУЧЕНИЯ ИНОСТРАННОГО ЯЗЫКА

Одной из самых важных составляющих изучения языка является изучение лексики. Наибольшей проблемой при изучении лексики является подбор слов для изучения. Подбирать слова можно различными способами, но это всегда ручная работа. Ее можно переложить на преподавателя или разработчика некоторой системы, но в конечном итоге эту работу делает человек.

В современном мире проводится много исследований как автоматизировать ту или иную работу, которую выполняет человек. Это уменьшает количество ошибок, улучшает производительность, а значит и качество выполненной работы. Таким образом, в данной работе предла-

гается переложить функцию подбора слов для изучения на программную систему. Система, которая определяет набор слов для изучения, относится к адаптивным системам обучения иностранному языку. Данная система позволит самостоятельно определять слова, необходимые пользователю для изучения, что сильно облегчит задачу поиска и изучения слов. Система решает эту задачу на основе индивидуальных характеристик обучаемого, таких, как количество изучаемых слов за определенный период времени, время, затрачиваемое на изучение одного слова, возраст и другие. Система, на основе собственных знаний, определит какие слова наиболее важны и предложит их для изучения.

Механизм рекомендации слов осуществляется путем предложения слов, изучаемых другими пользователями с похожими индивидуальными характеристиками. Для того, чтобы найти таких пользователей, предлагается использовать методы кластерного анализа. У каждого пользователя будет выделяться набор метрик, по которым будут формироваться кластеры пользователей (см. рис. 1), то есть пользователей с похожими индивидуальными характеристиками.

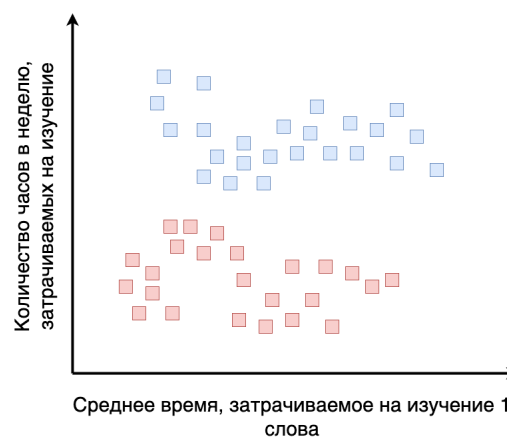


Рис. 1 – Пример выделения кластеров на основе индивидуальных характеристик.

К преимуществам данного подхода можно отнести то, что система автоматически подбирает слова для изучения, учитывая индивидуальные характеристики человека, тем самым повышая эффективность обучения.

III. СПИСОК ЛИТЕРАТУРЫ

1. Busuu Ltd [Электронный ресурс]. – Минск, 2020. – Режим доступа: <https://www.busuu.com/en/about> – Дата доступа: 04.10.2020.
2. Lingualo [Электронный ресурс]. – Минск, 2020. – Режим доступа: <https://lingualo.com> – Дата доступа: 04.10.2020.
3. Duolingo [Электронный ресурс]. – Минск, 2020. – Режим доступа: <https://www.duolingo.com> – Дата доступа: 04.10.2020.
4. Rosetta Stone Inc [Электронный ресурс]. – Минск, 2020. – Режим доступа: <https://www.rosettastone.com> – Дата доступа: 04.10.2020.

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ КОРОВНОЙ ФЕРМОЙ С БЛОКОМ ИДЕНТИФИКАЦИИ ЖИВОТНОГО И РАСПОЗНАВАНИЯ УРОВНЯ ЕГО АКТИВНОСТИ ПО ВИДЕО С WEB-КАМЕРЫ

Гунер М. В., Бусс И. А.

Кафедра информационных систем в экономике, Алтайский государственный технический университет имени И.И. Ползунова
Барнаул, Россия
E-mail: horyzont1@mail.ru

Блок распознавания активности позволит своевременно выявлять состояние животных. Данный блок реализован на языке программирования Python с использованием библиотеки компьютерного зрения OpenCV, библиотек машинного обучения и анализа данных tensorflow, keras, imageai и др.

Разработка автоматизированной информационной системы управления коровной фермой является актуальной, поскольку внедрение программы позволит руководителю более эффективно анализировать и контролировать работу предприятия. Система позволит сократить трудозатраты на поиск информации и формирование отчетов по деятельности фермы. Блок распознавания и оценки активности животных позволит своевременно выявлять состояние животных, что снизит риски несвоевременного обнаружения коров в состоянии «охоты», которые влекут за собой существенное снижение надоев молока, а впоследствии риск полной непригодности коровы для дальнейшего воспроизводства [1].

Цель работы – разработать систему управления коровной фермой с блоком идентификации животного и распознавания уровня его активности по видео с web-камеры.

Система предназначена для автоматизации управления содержанием животных, накопления и хранения информации о поголовье, его производительности и воспроизводстве. Система позволяет вести справочники животных, сотрудников, проводить документы отражающие надой, осеменения, фиксировать отелы, аборт и спускание животных с указанием причины, а также формировать стандартные отчеты для контролирующих ветеринарных служб, отчеты для внутреннего потребления, статистические отчеты.

Основная работа ведется с карточкой животного, в которой отображается вся информация о корове (рис. 1). На вкладке «Основное» мы можем посмотреть все данные о корове, год рождения, масть, породу, последний отел и т.д., а вкладка произведенные действия отражает все события, происходящие с животным: вакцинация, доение, осеменение и т.д.

Код:	000000010
Кличка (номер):	Клюква (69315426)
Кличка:	Клюква
Инвентарный номер:	69315426
Дата рождения:	13.02.2015
Дата последнего отела:	18.03.2020
Масть:	Темно-Красная
Порода:	Джерсийская
Дата вывоза:	03.06.2015
Место содержания:	Коровник №1
Статус воспроизводства:	Степная
Способ поступления:	Покупка
Мать:	Шима (66945654)
Отец:	Филька (26869603)

Рис. 1 – Карточка животного (Основное)

N	Дата	Тип события	Документ о корове	Описание
13	16.10.2017	ПУЛЬМОВ...		Вакцинация живот
14	06.12.2017	Запуск	Запуск 000000004 от 06.12.2017 12:00...	Запуск коровы в с
15	21.09.2018	ЛТФ-130		Вакцинация живот
16	14.05.2019	ПУЛЬМОВ...		Вакцинация живот
17	28.05.2020	Доение	Доение 000000008 от 28.05.2020 12:0...	Доение коровы
18	29.05.2020	Доение	Доение 000000009 от 29.05.2020 12:0...	Доение коровы
19	30.05.2020	Доение	Доение 000000010 от 30.05.2020 12:0...	Доение коровы
20	01.06.2020	Доение	Доение 000000002 от 01.06.2020 12:0...	Доение коровы
21	02.06.2020	Плыва	Плыва 000000003 от 02.06.2020 12:0...	Плыва коровы

Рис. 2 – Карточка животного (Произведенные действия)

Блок распознавания коров реализован на языке программирования Python с использованием библиотеки компьютерного зрения OpenCV, библиотек машинного обучения и анализа данных tensorflow, keras, imageai и др. Модель обнаружения объектов (RetinaNet, resnet50_coco_best_v2.0.1.h5), основанная на

сверточных нейронных сетях и поддерживаемая ImageAI, может обнаруживать 80 различных типов объектов. В настоящей работе нас интересовало обнаружение на изображении коров (тип объекта: cow) [2, 3,4].

Для идентификации коровы использовался механизм свертки изображений, т.е. обработки растровых изображений путем наложения на них фильтров, отвечающих за те или иные признаки [2,4]. Мы использовали 4 фильтра, кодирующие признак «цветовой окрас коровы», каждый фильтр представлял собой файл .jpg разрешением 7 пикселей * 7 пикселей (рис. 2) [4].

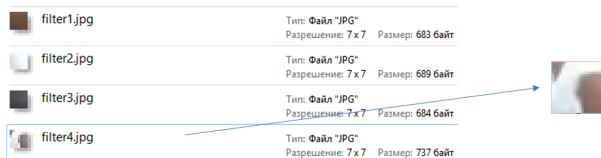


Рис. 3 – Фильтры, используемые при свертке изображений коров

Видеопоток, поступающий с веб-камеры, разбивается на кадры. Каждый кадр анализируется. С использованием механизмов идентификации объектов и слежения программа каждые 5 секунд подсчитывает количество смещений обнаруженных объектов в пространстве, в нашем случае коров, и строит график, иллюстрирующий активность того или иного животного. Работа системы распознавания и идентификации коров показана на рис. 3 и 4.

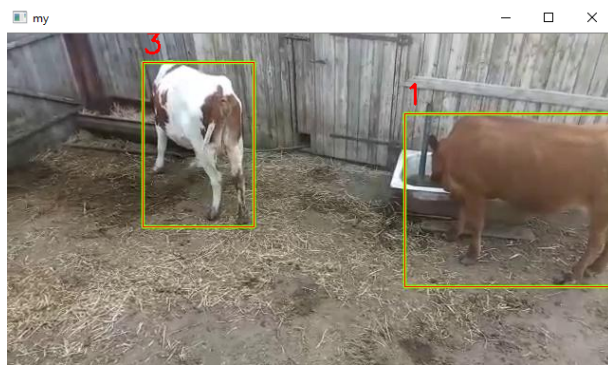


Рис. 4 – Результат работы подсистемы распознавания и идентификации коров

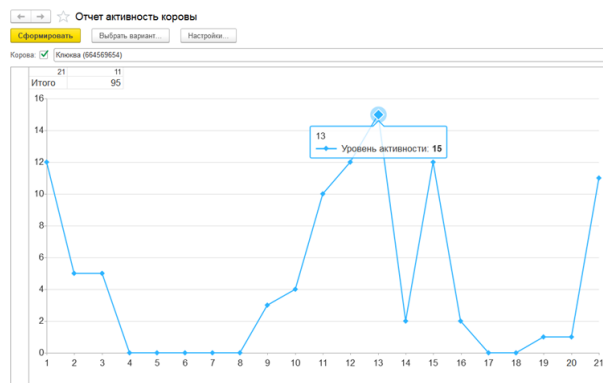


Рис. 5 – Результат работы подсистемы определения уровня их активности

Тестирование системы показало, что примененные в работе алгоритмы для распознавания и идентификации животного достаточно эффективные. Точность обнаружения и идентификации коровы составила 92 %. Однако, работу следует продолжать, так как окрас одной коровы от окраса другой коровы может отличаться весьма незначительно, и необходим более глубокий анализ «рубашек» коров.

СПИСОК ЛИТЕРАТУРЫ

1. Пути интенсификации воспроизводства стада в скотоводстве / А.В. Глаз и др. – Гродно: ГГАУ, 2011. – 80 с.
2. Нейронные сети и компьютерное зрение. Сверточные нейронные сети. Режим доступа: <https://stepik.org/course/50352/syllabus>
3. ImageAI. Detection Classes [Электрон.]. Режим доступа: <https://imageai.readthedocs.io/en/latest/detection/>
4. М.В. Гунер, И.А. Бусс разработка системы распознавания коров и определения уровня их активности на основе применения сверточных нейронных сетей // Нейроинформатика, её приложения и анализ данных: Материалы двадцать восьмого всероссийского семинара 25 сентября – 27 сентября 2020 года.: Красноярск 2020.– С. 38-41.

ФОРМУЛИРОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПРИЛОЖЕНИЙ, РАБОТАЮЩИХ С БОЛЬШИМИ ДАННЫМИ

Жук А. А.

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: san91130324@gmail.com

Рассматриваются проблемы постановки задачи оптимизации приложений, работающих с большими данными, специфика конвейеров обработки данных их типы, формулируются критерии оптимальности для задачи оптимизации приложений, работающих с большими данными

ВВЕДЕНИЕ

В связи с ростом количества приложений и технологий, работающих с большими данными, возникает потребность в решении задачи оптимизации данных приложений. Формулировка данной задачи и ее анализ является первым и важным шагом к построению эффективных приложений, работающих с большими данными

I. КОНВЕЙЕР ОБРАБОТКИ ДАННЫХ

В общем случае приложение, работающее с большими данными, можно представить как конвейер обработки данных, состоящий из блоков манипуляции (чтение, запись, трансформация) над данными. Блоки манипуляции могут выполняться как строго последовательно (см. рис. 1) так и параллельно (см. рис. 2).

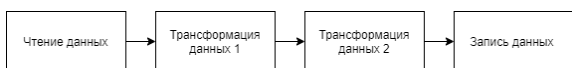


Рис. 1 – Пример конвейера обработки данных с последовательными блоками



Рис. 2 – Пример конвейера обработки данных, имеющего блоки, которые могут выполняться параллельно

Существует 2 типа конвейеров обработки данных:

- пакетные;
- потоковые.

Пакетные конвейеры обработки данных. Данный тип конвейеров обработки данных характеризуется тем, что работает с уже заранее готовыми наборами данных, хранящихся в определенных хранилищах данных.

Потоковые конвейеры обработки данных. Данный тип конвейеров обработки данных характеризуется тем, что работает с потоком данных в режиме реального времени.

II. КРИТЕРИЙ ОПТИМАЛЬНОСТИ

В рамках задачи оптимизации приложений, работающих с большими данными можно выделить 2 критерия оптимальности: время обработки данных и затраченные ресурсы на обработку данных. Для разных типов конвейеров обработки данных эти критерии оптимальности будут формулироваться следующим образом.

Для пакетного конвейера обработки данных:

- время выполнения приложения;
- затраченные ресурсы на выполнение приложения.

Для потокового конвейера обработки данных:

- время обработки заранее определенного объема данных;
- затраченные ресурсы на обработку заранее определенного объема данных.

III. ФОРМУЛИРОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ДЛЯ ПАКЕТНОГО КОНВЕЙЕРА ОБРАБОТКИ ДАННЫХ

Из всего выше описанного можно обозначить, что существуют функции f_t и f_r , где f_t - функция зависимости времени выполнения конвейера обработки данных от входных параметров, вариантов реализации блоков этого конвейера и объема данных, которого необходимо обработать, а f_r - функция зависимости затраты ресурсов на выполнения конвейера обработки данных от входных параметров, вариантов реализации блоков этого конвейера и объема данных, которого необходимо обработать. Тогда можно обозначить и функции f_i и g_i , где f_i - функция зависимости времени выполнения i -го блока конвейера обработки данных от входных параметров, вариантов реализации блоков этого блока и объема данных, которого необходимо обработать, а g_i - функция зависимости затраты ресурсов на выполнения i -го блока конвейера обработки данных от входных параметров, вариантов реализации блоков этого блока и объема данных, которого необходимо обработать. Функция f_i и g_i будут равны:

$$f_i = f_i(\lambda_i, N_i, x_1, x_2, \dots, x_k) \quad (1)$$

$$g_i = g_i(\lambda_i, N_i, x_1, x_2, \dots, x_k) \quad (2)$$

λ_i - это объем данных, который необходимо обработать i -му блоку конвейера обработки данных, N_i - это вариант реализации i -го блока конвейера обработки данных $N_i \in \{N_{i1}, N_{i2}, \dots, N_{iz}\}$, x_1, x_2, \dots, x_k - параметры конвейера обработки данных (количество узлов кластера хранения данных, количество оперативной памяти в кластере обработки данных и т.п.).

Тогда функция f_t принимает вид:

$$\begin{aligned} f_t = \phi(f_1(\lambda_1, N_1, x_1, x_2, \dots, x_k), \\ f_2(\lambda_2, N_2, x_1, x_2, \dots, x_k), \dots, \\ f_n(\lambda_n, N_n, x_1, x_2, \dots, x_k)) \end{aligned} \quad (3)$$

ϕ - это функция суммирования значений функций f_i , n - это количество блоков обработки данных.

В случае конвейера обработки данных с последовательными блоками функция f_t принимает вид:

$$f_t = \sum_{i=1}^n f_i(\lambda_i, N_i, x_1, x_2, \dots, x_k) \quad (4)$$

Функция f_r принимает вид:

$$\begin{aligned} f_r = \omega(g_1(\lambda_1, N_1, x_1, x_2, \dots, x_k), \\ g_2(\lambda_2, N_2, x_1, x_2, \dots, x_k), \dots, \\ g_n(\lambda_n, N_n, x_1, x_2, \dots, x_k)) \end{aligned} \quad (5)$$

ω - это функция суммирования значений функций g_i , n - это количество параметров конвейера обработки данных. В большинстве случаев функция ω представляет собой сумму значений функций g_i и принимает вид:

$$f_r = \sum_{i=1}^n g_i(\lambda_i, N_i, x_1, x_2, \dots, x_k) \quad (6)$$

Поскольку значения функций f_t и f_r имеют обратную зависимость и не могут в общем случае в задаче оптимизации рассматриваться независимо друг от друга, поэтому вводится функция β - функция сведения значений функций f_t и f_r к некоторому новому критерию оптимальности конвейера обработки данных. Тогда функция приобретает вид:

$$\begin{aligned} \beta = \beta(f_t(\lambda_1, \lambda_2, \dots, \lambda_n, N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k), \\ f_r(\lambda_1, \lambda_2, \dots, \lambda_n, N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k)) \end{aligned} \quad (7)$$

В этом случае возможно сформулировать задачу безусловной оптимизации (формула 8)

$$\min[\beta(\lambda_1, \lambda_2, \dots, \lambda_n, N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k)] \quad (8)$$

Задача конкретизации функции β в большинстве случаев лежит на эксперте предметной

области, в рамках которой разрабатывается конвейер обработки данных, так как именно он обладает знанием о том сколько ресурсов можно потратить и за какое время надо обработать данные. В случае когда функция β не может быть четко определена, можно сформулировать задачу условной оптимизации (условие 9)

$$\begin{cases} f_t = f_t(\lambda_1, \lambda_2, \dots, \lambda_n, \\ N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k) \leq T \\ f_r = f_r(\lambda_1, \lambda_2, \dots, \lambda_n, \\ N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k) \leq R \end{cases} \quad (9)$$

T - это максимально допустимое значение времени выполнения конвейера обработки данных, R - это максимально допустимое значение затрат ресурсов на выполнение конвейера обработки данных. Данные параметры формулируются экспертом предметной области, в рамках которой разрабатывается конвейер, поэтому нет гарантий, что при заданных T и R существует такие параметры $N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k$ при которых условие 9 будет выполняться. В случае если решения не существует, то надо пересмотреть значения T и R , если пересмотр значений T и R не привел к решению поставленной задачи, то можно сделать вывод о том что при текущем техническом оснащении невозможно удовлетворить потребность, которую призвано решить разрабатываемое приложение.

IV. ФОРМУЛИРОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ДЛЯ ПОТОКОВОГО КОНВЕЙЕРА ОБРАБОТКИ ДАННЫХ

Аналогично, как и для задачи оптимизации пакетного конвейера обработки данных, формулируется задача для потокового конвейера обработки данные, единственным отличием для потоковых конвейеров является то, что функции f_t и f_r базируются не на времени и затратах на выполнения конвейера обработки данных, а времени и затратах на обработку некоторого заранее определенного объема данных.

V. ВЫВОДЫ

Из сформулированной задачи оптимизации конвейера обработки данных можно сделать вывод о том, что качество оптимизации конвейера обработки данных зависит от того насколько была проделанная работа по определению значений T и R , и по поиску всех возможных значений параметров $N_1, N_2, \dots, N_n, x_1, x_2, \dots, x_k$. Так же стоит отметить что приложение оптимальное для обработки объема данных λ_1 может быть не оптимальным для обработки данных объема λ_2 , поэтому всегда стоит заранее обсуждать вопрос текущего объема данных и то до какого объема он может вырасти с экспертом предметной области.

1. Дэн Саймон. Алгоритмы эволюционной оптимизации. / Дэн Саймон // 2020. - 1002 с.

ИСПОЛЬЗОВАНИЕ АСТОР-CRITIC АЛГОРИТМОВ ПРИ ОБУЧЕНИИ АГЕНТОВ ДЛЯ ИГР НА ATARI 2600

Зязюлькин С. П., Нестеренков С. Н.

Факультет компьютерных систем и сетей, кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: sergeyzyazyulkin@gmail.com, nsn@bsuir.by

Последние достижения в области машинного обучения с подкреплением позволяют обучать агентов, значительно превосходящих человека в решении некоторых задач. Одним из самых популярных бенчмарков для алгоритмов и моделей машинного обучения с подкреплением являются игры на Atari 2600. В данной статье рассматриваются actor-critic алгоритмы и их модификации при обучении агентов для игр на Atari 2600.

ВВЕДЕНИЕ

Агент представляет собой сущность, которая получает на вход наблюдение (состояние) окружающей среды s и выполняет некоторое действие a , определяемое политикой π , с целью максимизации суммарной награды. Существует два больших семейства алгоритмов машинного обучения с подкреплением: DQN [1] и Policy Gradient [2].

DQN алгоритмы предполагают приближение нейронной сетью Q-функции, определяющей ценность действия a при наблюдении s :

$$Q(s, a) = E_{s' \sim P(\cdot|s, a)} \left[R(s, a) + \gamma \max_{a' \in A} Q(s', a') \right],$$

где A – множество допустимых действий, R – награда за выполнение действия a в состоянии s , γ – дисконтирующий множитель.

Политика агента в этом случае является детерминированной и имеет следующий вид:

$$\pi(s) = \operatorname{argmax}_a Q(s, a).$$

Policy Gradient алгоритмы, в отличие от DQN, предполагают приближение нейронной сетью непосредственно самой политики π . Во-первых, это позволяет избежать необходимости решать задачу оптимизации – находить действие, имеющее максимальную ценность в данном состоянии. Для сред с небольшим числом допустимых действий решение задачи оптимизации не является проблемой, однако задача оптимизации может стать очень сложной в случае, когда число допустимых действий большое или действие является непрерывным, т.е. имеет бесконечное число допустимых значений. Во-вторых, Policy Gradient позволяет обучать стохастические политики, когда в DQN предпринимаемое действие является детерминированным для каждого состояния.

Градиент политики определяется следующей формулой:

$$\nabla J \approx E [Q(s, a) \nabla \log \pi(a|s)].$$

С практической точки зрения оптимизация градиента политики может быть реализована как оптимизация функции потерь вида

$$L = -Q(s, a) \log \pi(a|s).$$

Значение функции $Q(s, a)$ может быть разбито на две части: ценность состояния $V(s)$ и преимущество $A(s, a)$, которое даёт действие a . Использование $A(s, a)$ вместо $Q(s, a)$ в градиенте политики позволяет существенно повысить стабильность обучения. Для вычисления $V(s)$ используется отдельная нейронная сеть. Нейронная сеть, отвечающая за вероятности действий, называется actor, отвечающая за ценность состояния – critic. Этот алгоритм имеет название advantage actor-critic или A2C.

МОДИФИКАЦИИ АСТОР-CRITIC АЛГОРИТМОВ

Для улучшения исследования среды в Policy Gradient алгоритмах часто добавляется entropy loss:

$$L_H = - \sum_i \pi_\theta(s_i) \log \pi_\theta(s_i).$$

Entropy loss принимает минимальное значение в случае равномерного распределения вероятностей. Использование entropy loss позволяет избегать политик, в которых одно действие слишком сильно доминирует над остальными.

Алгоритм A2C допускает эффективное распараллеливание. Создаётся несколько процессов, каждый из которых взаимодействует с одной или несколькими средами и генерирует переходы (s, a, r, s') . Все переходы собираются на одном процессе, который вычисляет функцию потерь и выполняет шаги стохастического градиентного спуска. Обновлённые веса нейронной сети передаются на все процессы, выполняющие взаимодействие со средой. Параллельный вариант алгоритма A2C с асинхронным взаимодействием со средами получил название asynchronous advantage actor-critic или A3C [3].

Дальнейшим улучшением A2C является алгоритм PPO, предложенный командой OpenAI в

2017-м году [4]. Основным предлагаемым алгоритмом улучшением является изменение формулы градиента политики. Градиент политики вычисляется как отношение новой и старой политик, масштабируемое значением преимущества действия:

$$J_{\theta} = E_t \left[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t \right].$$

Использование такого значения градиента политики может приводить к очень большим изменениям в весах нейронной сети, что может нарушить весь процесс обучения, т.к. данные для дальнейшего обучения генерируются текущей политикой, т.е. ухудшение политики приводит к снижению качества генерируемых данных. Для решения этой проблемы величина отношения новой и старой политик ограничивается интервалом $[1 - \epsilon, 1 + \epsilon]$. Варьирование величины ϵ позволяет ограничить величину изменения весов.

Ещё одно отличие от A2C – способ приближения величины преимущества действия. В A2C преимущество вычисляется по формуле

$$A_t = -V(s_t) + r_t + \dots + \gamma^{T-t-1} r_{T-1} + \gamma^{T-t} V(s_T).$$

В PPO используется более общее приближение преимущества:

$$A_t = \sigma_t + (\gamma\lambda)\sigma_{t+1} + \dots + (\gamma\lambda)^{T-t-1}\sigma_{T-1},$$

где $\sigma_t = r_t + \gamma V(s_{t+1}) - V(s_t)$.

Приближение преимущества в A2C является частным случаем приближения преимущества в PPO при $\lambda = 1$.

Также стоит отметить, что в PPO имеются отличия в процессе обучения: генерируется длинная последовательность переходов, после чего подсчитываются преимущества действий для всей последовательности и выполняется несколько эпох обучения.

Одной из больших проблем машинного обучения с подкреплением является процесс исследования среды во время обучения. Этой проблеме посвящена работа [5]. В ней предложено использовать дополнительную награду за «любопытство». Это позволяет улучшить процесс исследования среды в процессе обучения и, как следствие, улучшить качество итоговых агентов, когда награда является разреженной. Награда за «любопытство» вычисляется при помощи двух нейронных сетей со схожей архитектурой. Одна из сетей, именуемая целевой, инициализируется случайными весами. Эти веса остаются неизменными на всём процессе обучения. Другая сеть, именуемая предсказывающей, обучается предсказывать выходы целевой нейронной сети. На вход сетям подаётся итоговое состояние среды на

текущем шаге. Для вычисления величины награды за «любопытство» используется следующая формула:

$$r_t^i = \left\| \hat{f}(s_{t+1}) - f(s_{t+1}) \right\|_2^2,$$

где \hat{f} и f – предсказывающая и целевая сети соответственно. Данный подход получил название RND (Random Network Distillation).

Рассмотрим способы повышения скорости обучения при обучении агентов для игр на Atari 2600. На вход игровому агенту подаётся несколько цветных кадров игрового пространства с разрешением 210x160. Преобразование игровых кадров в чёрно-белые изображения меньшего разрешения позволяет значительно уменьшить размер кадра. Так, например, сжатие кадра до чёрно-белого изображения с разрешением 80x80 позволяет уменьшить размер почти в 16 раз, что значительно сказывается на скорости обучения, при этом обычно отсутствует заметное влияние на качество итогового игрового агента. Другой возможной оптимизацией является применение выбранного игровым агентом действия для нескольких последовательных взаимодействий со средой. Если частота взаимодействий со средой велика и агент часто предпринимает одно и то же действие для последовательных взаимодействий, имеется возможность «сэкономить» на числе взаимодействий агента без потери качества, особенно в условиях, когда обработка состояния среды игровым агентом является «тяжёлой» операцией.

ЗАКЛЮЧЕНИЕ

Рассмотренные actor-critic алгоритмы и их модификации позволяют обучать игровых агентов, превосходящих человека в большом числе игр на Atari 2600. Изучение влияния отдельных модификаций и их комбинаций на обучение игровых агентов для конкретных игр на Atari 2600, а также поиск иных модификаций остаются открытыми вопросами для дальнейшего изучения.

1. Mnih, V. Playing Atari with Deep Reinforcement Learning / V. Mnih, K. Kavukcuoglu, D. Silver, [и др.] // arXiv:1312.5602.
2. Thomas, P. Policy Gradient Methods for Reinforcement Learning with Function Approximation and Action-Dependent Baseline / P. Thomas, E. Brunskill // arXiv:1706.06643.
3. Mnih, V. Asynchronous Methods for Deep Reinforcement Learning / V. Mnih, M. Mehdi, A. Graves, [и др.] // arXiv:1602.01783.
4. Schulman, J. Proximal Policy Optimization Algorithms / J. Schulman, F. Wolski, P. Dhariwal, [и др.] // arXiv:1707.06347.
5. Burda, Y. Exploration by Random Network Distillation / Y. Burda, H. Edwards, A. Storkey, O. Klimov // arXiv:1810.12894.

ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РЕАЛИЗАЦИИ ПАРАЛЛЕЛЬНОЙ РЕДУКЦИОННОЙ СХЕМЫ ТЕОРЕТИКО-МНОЖЕСТВЕННЫХ ОПЕРАЦИЙ

Ивашенко В. П.

Кафедра интеллектуальных информационных технологий, Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: ivashenko@bsuir.by

В работе приведены результаты исследований для обработки информации на уровне управления знаниями, представленными средствами модели унифицированного семантического представления знаний, особенностью которой является наличие базовой теоретико-множественной семантики представленных знаний.

ВВЕДЕНИЕ

В технических системах, способных к обучению и ориентированных на решение задач, к которым относятся системы, управляемые знаниями (СУЗ) [1], существенным является качество масштабируемости [2] за счёт соблюдения требований по ключевым качествам, включая производительность. С целью обеспечения соблюдения этих требований предлагаются различные модели обработки информации для разных уровней управления СУЗ [1], [3].

I. РЕДУКЦИОННАЯ СХЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ

Рассмотрим задачу, удовлетворяющую двум требованиям: 1) входными данными задачи являются линейно упорядоченные множества данных (массивы); 2) её решение сводится к пошагово-последовательному чтению данных последовательно из ячеек этих массивов и возможной записи в качестве выходных данных результата, вычисленного на основе прочитанных на текущем шаге данных и данных, полученных в результате выполнения операции редукции (ОР) – бинарной ассоциативной операции, применённой к прочитанным данным текущего шага и аналогичному результату ОР предыдущего шага. Следовательно, решение задачи может быть разбито на две фазы вычислений: результата ОР и окончательного результата. Очевидно, что задача в фазе 2 является частным случаем исходной задачи для одноэлементных массивов входных данных. С целью параллельной обработки массивы разбиваются на части. Чтобы не хранить результаты ОР для каждого элемента массивов, предлагается хранить результат ОР только для начальных элементов частей массивов, а для всех последующих элементов вычислять результат ОР совместно с вычислением окончательного результата, т. е. решать частный случай исходной задачи целиком. Для первых частей массивов решение задачи может быть начато сразу, без решения задачи фазы 1, так как

не требуется расчёт результата ОР ввиду отсутствия предшествующих элементов, также не требуется расчёт результата ОР – решение задачи фазы 1 – для последних частей массивов, так как после них нет каких-либо частей, которые бы требовали этот результат.

Коэффициент ускорения (КУ) решения задачи на p процессорах можно выразить как $(k + p)/(k + 1)$, при этом исходные массивы разбиваются на $p + 1$ часть, а исходная задача сводится к задачам обработки этих частей; полагая, что отношение времени решения задачи фазы 1 ко времени решения всей задачи составляет k , то, чтобы выравнять время решения всех параллельно решаемых задач на первом этапе, первая часть в k раз короче остальных. Соответствующий график для $k = 1$ представлен на рисунке 1, а редукционная схема (РС) – на рисунке 4 (светло-серым показана фаза 1 решения задачи, а тёмно-серым – решения остальных задач для частей).

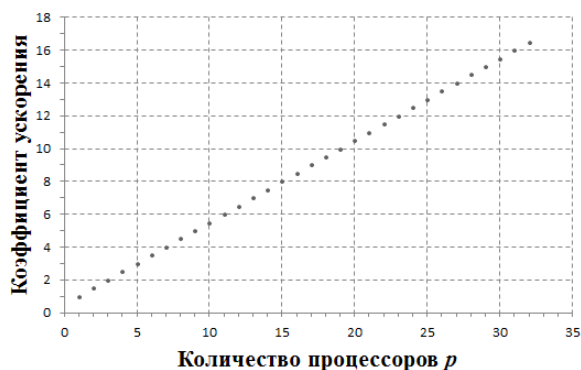


Рис. 1 – Теоретическая зависимость коэффициента ускорения от количества процессоров

II. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для экспериментов были выбраны задачи вычисления результатов операций пересечения, объединения и симметрической разности множеств, заданных парой массивов с n и m 32-битными элементами ($n \leq m$). Тестирование проводилось на архитектуре AMD Ryzen Threadripper 2950X (8Гиб ОЗУ, Windows 10 x64)

для размеров множеств от 1 до 10^8 . В качестве последовательного решения были взяты алгоритмы, обеспечивающие временные сложности решения задач: $O(n * \ln(m/n) * f(n + m))$ – для пересечения, $O((n+m) * f(n+m))$ – для объединения и симметрической разности, где $f(x)$ – время доступа к элементу массива размера x . Элементами для пар множеств брались натуральные числа случайно из диапазона от 1 до $2 * (n + m)$, а также – комбинации чётных, нечётных и т. п. наборов чисел из этого диапазона, чисел от его начала и с конца, одинаковые наборы и наборы одинаковых чисел. Для разбиения на части применялся подход MergePath [4]. Графики результатов экспериментов приведены на рисунках 2 и 3.

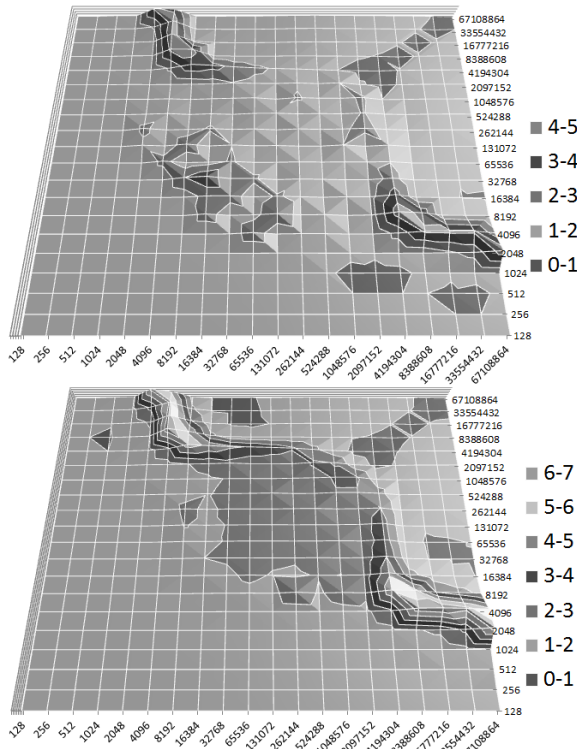


Рис. 2 – КУ пересечения мультимножеств для 4х (сверху) и 8и (снизу) потоков

Аномально высокие значения КУ для операции пересечения можно объяснить особенностью работы кэш-памяти и спецификой задачи пересечения мультимножеств. В остальном полученные значения КУ соответствуют теоретическим. Влияние архитектуры существенно: выигрыш от применения параллельной РС отсутствует или незначителен для множеств, ёмкость которых не превышает 32КиБ (размер кэша первого уровня) либо превышает 16МиБ (размер кэ-

ша третьего уровня на пару ССХ). Увеличение количества потоков более 8и не даёт существенного прироста в силу особенностей (NUMA) архитектуры AMD Ryzen Threadripper 2950X.

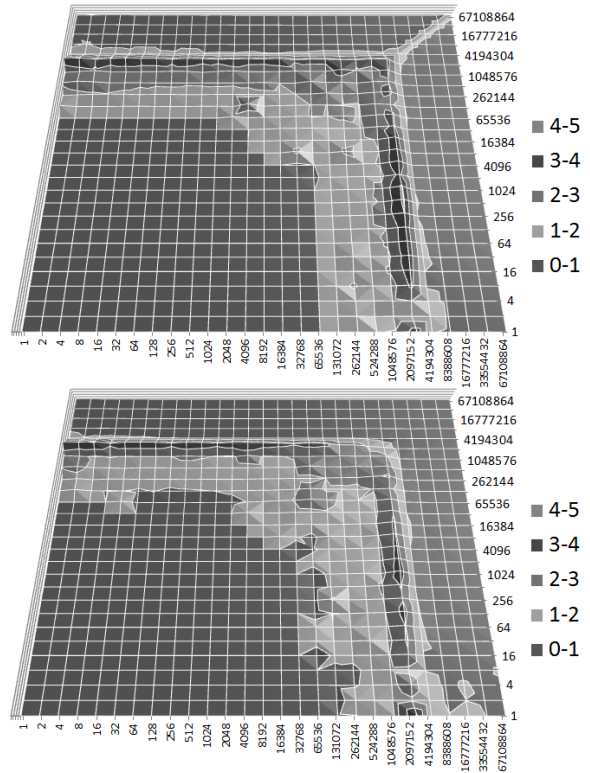


Рис. 3 – КУ симметрической разности (сверху) и объединения (снизу) мультимножеств (8 потоков)

ЗАКЛЮЧЕНИЕ

Эксперименты показали работоспособность предложенной РС, её согласованность с теоретическими оценками и возможность её применения для повышения производительности операций обработки знаний в СУЗ.

1. Ивашенко, В. П. Операции управления массивами данных в линейно адресуемой памяти / В. П. Ивашенко, С. В. Синцов // Доклады БГУИР. – Vol. 100, №6, – Минск, 2016. – С. 86–93.
2. Hill, M. D. What is scalability? / M. D. Hill // ACM SIGARCH Computer Architecture News. – Vol. 18, №4. – 1990. – pp. 18–21.
3. Ivashenko, V. P. String processing model for knowledge-driven systems / V. P. Ivashenko // Doklady BGUIR. – Vol. 18, №6, – Минск, 2020. – pp. 33–40.
4. Green, O. GPU merge path: A GPU merging algorithm / O. Green, R. McColl, D. A. Bader // Proceedings of the 26th ACM International Conference on Supercomputing, ser. ICS '12. – NY, USA: ACM, 2012. –pp. 331–340.

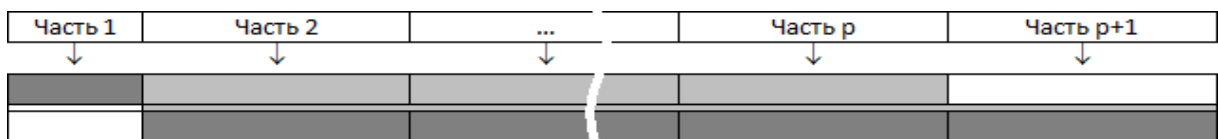


Рис. 4 – Редукционная схема решения задачи

ОПЕРАЦИОННАЯ СЕМАНТИКА МНОГОАГЕНТНЫХ СИСТЕМ ОБРАБОТКИ ЗНАНИЙ

Ивашенко В. П.

Кафедра интеллектуальных информационных технологий, Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: ivashenko@bsuir.by

В работе исследуются виды операций, их связь с состояниями и элементами состояний, рассматриваемыми как информационные конструкции, в рамках моделей обработки знаний

ВВЕДЕНИЕ

В системах, управляемых знаниями [1], ориентированных на решение задач и способных к обучению, важным качеством является способность адекватно представлять явления и процессы становления онтологических структур [1,2], приобретения знаний, взаимодействия с внешней средой.

I. ЗАДАЧИ ОБРАБОТКИ ЗНАНИЙ ДЛЯ АГЕНТОВ В СИСТЕМАХ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Пусть все процессы обработки информации в системе, управляемой знаниями, на уровне управления знаниями задаются моделью обработки знаний [2]: $\langle A, L, R, B, I, O, \mu \rangle$, где A – алфавит формального языка L , R – его множество синтаксических отношений, B – множество начальных информационных конструкций ($B \subseteq L$), I – множество интерпретаций текстов формального языка L , O – множество операций, μ – семантическая метрика. Каждая операция понимается как бинарное отношение ($O \subseteq 2^{M \times M}$) на множестве всех состояний (ситуаций) M , которое является подмножеством L ($B \subseteq M$). Операция называется детерминированной, если и только если она является функцией (однозначным соответствием), иначе операция – недетерминированная. Операция называется обратимой, если и только если обратное ей отношение является детерминированной операцией, иначе операция называется необратимой. Операция называется возвратной операцией, если и только если её транзитивное замыкание является симметричным бинарным отношением. Любая индивидуальная задача (задача с единственной исходной ситуацией) на множестве ситуаций M , может быть задана операцией из $2^{M \times M}$. Задача (индивидуальная или обобщённая) называется задачей познания, если и только если некоторые составляющие её индивидуальные задачи задаются недетерминированными операциями, иначе задача называется задачей исполнения. На основе введённых понятий может быть задана общая классификация задач и соответствующих операций (см. таблицу 1).

Индивидуальные задачи могут быть только задачами поиска, конструкции и реконструкции.

Таблица 1 – Общая классификация задач

Задачи	Обратимые		Необратимые	
	Невозвратные	Возвратные	Невозвратные	Возвратные
Познания	Поиск	–	Выбор	Проверка
Исполнения	Конструкция	Реконструкция	–	Дефекструкция

Представление и обработка знаний рассматриваются как явления их бытия и изменения, являющихся частными видами становления (операции) [1]. Становлением актуального и неактуального [1,3] наряду с процессами обработки знаний задаются процессы приобретения знаний, относящиеся к процессам познания. Так, множество, может быть задано процессом кодирования – становлением знака множества из его элементов, и процессом декодирования – становлением элементов множества из его знака, а множество состояний может быть задано их бытием (становлением). Пусть объединение всех операций есть Γ , а множество целевых состояний (для игрока в игре с полной информацией [4]) задано операцией становления τ , тогда область объективных гарантий τ есть

$$\bigcap_{k=1}^{<\omega} \bigcup_{i=k}^{<\omega} \Phi^i(\tau)$$

где

$$\begin{aligned} \Phi^1 &= \Phi, & \Phi^i &= \Phi^{i-1} \circ \Phi, \\ \Phi(\tau) &= \tau \cup \Gamma^{\pm} \tau, \\ \Gamma^{\pm} \tau &= \tau \overset{\leftarrow}{\circ} (\Gamma^{-1}), \\ \alpha \overset{\leftarrow}{\circ} \beta &= (\alpha \circ \beta) \cap (\alpha \overset{\leftarrow}{\bullet} \beta), \\ \alpha \overset{\leftarrow}{\bullet} \beta &= M \times M - ((M \times M - \alpha) \circ \beta), \\ \alpha \overset{\rightarrow}{\bullet} \beta &= M \times M - \alpha \circ ((M \times M - \beta)). \end{aligned}$$

Здесь ω – ординал, а $\overset{\leftarrow}{\circ}$, $\overset{\leftarrow}{\bullet}$, $\overset{\rightarrow}{\bullet}$ – (мета)операции соответственно: индуктивной композиции (демонической композиции [5]), индукции и кондукции.

Если соотнести операции с разными игроками, то каждая операция может быть рассмотрена как агент [4,6] в многоагентной системе [6].

Пусть ι – тождественная операция (бытия), и множество позиций (состояний), в которых игроку [7] принадлежит ход [4], задано операцией

бытия ψ ($\psi \subseteq \iota$), а множество позиций, в которых ход принадлежит соперникам, задано операцией бытия φ ($\varphi \cup \psi = \iota$), тогда область полусубъективных гарантий τ есть

$$\bigcap_{k=1}^{<\omega} \bigcup_{i=k}^{<\omega} \Phi_{\varphi^i}(\tau),$$

где

$$\begin{aligned} \Phi_{\varphi}(\tau) &= \tau \cup \Gamma_{\varphi}^{\pm} \tau \cup \Gamma_{\varphi}^{-} \tau, \\ \Gamma_{\varphi}^{\pm} \tau &= (\Gamma \cap (\varphi^{-1} \circ \emptyset))^{\pm} \tau, \\ \Gamma_{\varphi}^{-} \tau &= \tau \circ (\Gamma \cap (\varphi^{-1} \circ (M \times M)))^{-1}, \end{aligned}$$

а область субъективных гарантий [4] τ есть

$$\bigcap_{k=1}^{<\omega} \bigcup_{i=k}^{<\omega} \Phi_{\psi^i}(\tau),$$

где

$$\begin{aligned} \Phi_{\psi}(\tau) &= \tau \cup \Gamma_{\psi}^{\pm} \tau \cup \Gamma_{\psi}^{-} \tau, \\ \Gamma_{\psi}^{\pm} \tau &= (\Gamma \cap (\psi^{-1} \circ (M \times M)))^{\pm} \tau, \\ \Gamma_{\psi}^{-} \tau &= \tau \circ (\Gamma \cap (\psi^{-1} \circ \emptyset))^{-1}. \end{aligned}$$

II. КЛЮЧЕВЫЕ КОМПОНЕНТЫ И ЭЛЕМЕНТЫ ЯЗЫКОВ С ОПЕРАЦИОННОЙ СЕМАНТИКОЙ

Каждая операция обработки знаний Γ задаёт разбиения языка L и множества состояний M на подязыки в соответствии с выражением

$$D = \Delta(\langle \langle \langle \langle \langle M, L/M \rangle, B \rangle, G \rangle, \Gamma \rangle \rangle / \{ \emptyset, L \}$$

где

$$\begin{aligned} G &= \varepsilon((\Gamma \circ \Gamma^{-1}) / (\Gamma^{-1} \circ \Gamma)), \\ \varepsilon(\Gamma) &= \{ \chi \mid \langle \chi, \chi \rangle \in \Gamma^{-1} \circ \Gamma \}, \\ \Lambda(\langle \alpha, \beta \rangle) &= \{ \chi \cap \beta \mid \chi \in \alpha \} \cup \{ \chi \cap (L/\beta) \mid \chi \in \alpha \}, \\ A(\langle \alpha, \beta, \Gamma \rangle) &= \Lambda(\langle \langle \Lambda(\langle \alpha, \varepsilon(\beta \circ \Gamma) \rangle), \varepsilon(\beta) \rangle \rangle), \\ \Omega(\langle \beta, \Gamma \rangle) &= \Gamma / (\Gamma^{-1} \circ (\Gamma^{-1} \circ \beta)), \\ \Psi(\langle \alpha, \beta, \Gamma \rangle) &= \langle A(\langle \alpha, \beta, \Gamma \rangle), \Omega(\langle \beta, \Gamma \rangle) \rangle, \\ \Delta(\langle \alpha, \Gamma \rangle) &= \alpha \cap \bigcup_{\langle \chi, \gamma \rangle \in \Gamma} \Delta(\Psi(\langle \langle \alpha, \{ \chi \}^2 \circ \Gamma, \Gamma \rangle \rangle)) \end{aligned}$$

Наименьший набор ключевых элементов [8] соответствующих подязыков $N(P(D))$, задающих то же самое разбиение D , будем рассматривать как набор ключевых элементов, задающих операционную семантику и обладающих ею, где

$$\begin{aligned} B(\alpha) &= (\alpha^{\alpha} \cap \{ \delta^{-1} \mid \delta \in \alpha^{\alpha} \}) / \{ \langle \chi, \chi \rangle \mid \chi \in \alpha \}, \\ K(\delta) &= C(\langle \langle \bigcup_{\langle \chi, \gamma \rangle \in \delta} x, \bigcup_{\langle \chi, \gamma \rangle \in \delta} \gamma, A, T \rangle \rangle, \\ &\quad \Theta(\delta) = \{ \chi \cup \gamma \mid \langle \chi, \gamma \rangle \in \delta \}, \\ P(\alpha) &= \{ K(\delta) \cup \chi \mid (\chi \in P(\Theta(\delta))) \wedge (\delta \in B(\alpha)) \}, \\ Z(\alpha) &= \{ \chi \mid \exists \beta (L(\langle \alpha, \beta \rangle) \wedge (\chi \in \beta)) \}, \\ N(\alpha) &= \arg_{\chi \in Z(\alpha)} (\langle \langle \chi \mid \min(\{ \gamma \mid \gamma \in \alpha \}) \rangle \rangle), \end{aligned}$$

а C (функция ключевых элементов), T (предикат комбинаций разбиения) и L (функция наименее мощных множеств) наряду с \arg определены

в [8]. Если в модели обратить все процессы становления, т.е. обратить все операции (агенты), то полученные для такой модели (агента) ключевые элементы с операционной семантикой для исходной модели (агента) будем называть ключевыми элементами ко-операционной семантики.

Аналогичным образом могут быть определены ключевые компоненты, фразы и схемы [8], задающие операционную семантику.

ЗАКЛЮЧЕНИЕ

В соответствии с онтологической моделью пространственно-временных отношений событий и явлений для процессов обработки знаний [1] выделены наиболее общие виды задач и проведён анализ основных видов гарантий при их решении агентами обработки знаний в многоагентных системах, управляемых знаниями и соответствующих условиям игр с полной информацией. Предложен подход для выявления ключевых компонентов и элементов языка обработки знаний на основе сведения анализа его текстов, обладающих операционной семантикой, к анализу языков представления знаний.

- Ивашенко, В. П. Онтологическая модель пространственно-временных отношений событий и явлений в процессах обработки знаний / В. П. Ивашенко // Вестник БрГТУ. – 2017. – №5(107). – С. 13–17.
- Ivashenko, V. P. Semantic logging of knowledge processing based on binary generated events / V. P. Ivashenko // Pattern Recognition and Information Processing (PRIP 2019) : Proceedings of the 14th International Conference, Minsk, 21–23 May 2019, “Bestprint”, Belarus. – Minsk, 2019. – P. 172–177.
- Гегель, Г. В. Ф. Наука логики / Г. В. Ф. Гегель ; в 3-х тт. Т.1. : – Москва: Мысль. – 1970. – с. 501.
- Берж, К. Общая теория игр нескольких лиц / К. Берж ; Пер. с фр. : – М. : ФизМатЛит. – 1961. – с. 129.
- Backhouse, R. Demonic operators and monotype factors / R. Backhouse, J. van der Woude // Mathematical Structures in Computer Science. – 1993. – 3(4). – P. 17–433.
- Ивашенко, В. П. Применение мультиагентного подхода для реализации виртуальной графодинамической среды / В. П. Ивашенко, А. Л. Кондратенко // Четвертая международная летняя школа-семинар по искусственному интеллекту для студентов и аспирантов : сб. науч. тр., Республика Беларусь, Браславские озера, 29 июня – 6 июля 2000 г. / Белорус. гос. ун-т ; редкол. : О.П. Кузнецов (отв. ред.) [и др.]. – Минск, 2000. – С. 154–163.
- Blass, A. A game semantics for linear logic / A. Blass // Annals of Pure and Applied Logic. – 1992. – 56. – P. 151–166.
- Ivashenko, V. Attributes, scales and measures for knowledge representation and processing models / V. Ivashenko // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2019) : материалы международной научно-технической конференции, Минск, 21 – 23 февраля 2019 г. / Белорусский государственный университет информатики и радиоэлектроники; редкол.: В. В. Голеников (гл. ред.) [и др.]. – Минск, 2019. – С. 247–250.

СПРАВОЧНО-ПРОВЕРЯЮЩАЯ СИСТЕМА НА ОСНОВЕ УНИФИЦИРОВАННОГО СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Ивашенко В. П.

Кафедра интеллектуальных информационных технологий, Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: ivashenko@bsuir.by

В работе рассмотрено применение моделей обработки информации для систем, управляемых знаниями, и модели унифицированного семантического представления знаний для реализации вопросно-ответных систем на примере справочно-проверяющей системы

ВВЕДЕНИЕ

Для интеллектуальных обучающих систем [1], способных проверять ответы пользователя, одним из важных качеств является умение самостоятельно отвечать на вопросы и решать задачи, решение которых требуется от пользователя. Тогда как существование таких систем невозможно без базы необходимых знаний, их унифицированное семантическое представление способно облегчить интеграцию новых знаний и расширение класса решаемых задач в процессе обучения [1,2]. Системы, способные предоставлять информацию и проверять правильность ответов о предметной области, будем относить к справочно-проверяющим системам. Рассмотрим строение таких систем, как систем, управляемых знаниями [3].

1. ПЛАТФОРМА И КОМПОНЕНТЫ СПРАВОЧНО-ПРОВЕРЯЮЩЕЙ СИСТЕМЫ

Справочно-проверяющая система как система, основанная на знаниях, состоит из подсистем: пользовательского интерфейса, решателя задач и базы знаний [2].

В качестве платформы для реализации [4] была выбрана система из двух виртуальных машин: машины обработки образов и машины обработки онтологий (обработки знаний). Устройство этих машин соответствуют нижним уровням управления для систем, управляемых знаниями [3]. В частности, для машины обработки образов были разработаны системы команд, обеспечивающих решение задач и реализацию следующих операций [5]:

- логической 32-разрядной обработки данных [5];
- целочисленной 32-разрядной арифметики [5];
- работы с указателями на ячейки памяти и управления памятью и исполнением команд [3,6];
- арифметики для чисел с плавающей точкой двойной точности;
- алгебры векторов и матриц для чисел с плавающей точкой двойной точности;

- обработки строк и списковых структур данных [7];
- вывода данных, включая графические, текстовые и звуковые данные;
- ввода данных, включая операции управления и обработки событий получения новых данных.

Машина обработки образов имеет память с линейным адресным пространством и ориентирована на реализацию универсальных моделей решения задач и обработку сцен образов произвольной сложности [3,5,6]. Для реализации справочно-проверяющей системы использовалась web-реализация машины обработки образов.

Машина обработки онтологий является мультитековой виртуальной машиной и имеет модульную архитектуру и память с ассоциативным доступом к данным с ориентацией на обработку сложно структурированных данных и нечисловой информации на основе моделей унифицированного семантического представления знаний и обработки строк для систем, управляемых знаниями. Для машины обработки образов были разработаны системы команд, обеспечивающих решение задач и реализацию следующих операций:

- целочисленной 32-разрядной арифметики;
- нецелочисленной арифметики;
- обработки строк и списковых структур данных;
- работы со стеком и управления исполнением команд;
- обработки множеств и онтологических структур на основе становления актуального и неактуального [2];
- ввода-вывода данных.

К основным компонентам справочно-проверяющей относятся интерфейсные агенты (агенты ввода-вывода) и агенты обработки базы знаний. Первые реализованы средствами машины обработки образов, на базе объектно-ориентированного подхода, при этом агентно-ориентированный подход рассматривается как его развитие [8]. Вторые реализованы

средствами машины обработки онтологий, база знаний которых хранит: онтологию разделов предметной области, за которые они отвечают; естественно-языковые конструкции необходимые для описания содержимого этих разделов при взаимодействии с пользователем.

II. ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ СПРАВОЧНО-ПРОВЕРЯЮЩЕЙ СИСТЕМЫ

Протокол взаимодействия между виртуальными машинами реализован посредством протокола WebSocket. Интерфейсные агенты ввода-вывода (сенсорные и эффекторные агенты) получают данные сцены (в частности – текстовые данные) от агентов из машины обработки онтологии, обрабатывают их и выводят пользователю. При получении ответа, обработанный ответ вместе с данными о реакции пользователя, накопленными в процессе взаимодействия с ним отправляется к агентам в машину обработки онтологий, в которой множество агентов взаимодействуют через общую базу знаний, каждый отвечая за собственный раздел предметной области и её семантической окрестности [1,2,5]. Взаимодействие компонентов можно увидеть на схеме (см. Рис. 1).

ЗАКЛЮЧЕНИЕ

Разработанные прототипы справочно-проверяющей системы продемонстрировали работоспособность выбранных моделей и решений и возможность обеспечения снижения затрат на проведение тестирования уровня знаний учащихся при применении в учебном процессе вузов. Перспективой развития является совершенствование механизмов работы агентов с базой знаний, в том числе за счёт её расширения, и управления процессом взаимодействия с пользователем на основе семантического протоколирования [9] этого процесса.

1. Ivashenko, V. P. Semantic representation with knowledge measuring for distance learning / V. P. Ivashenko // Дистанционное обучение – образовательная среда XXI века : материалы XI

Международной научно-методической конференции, Минск, 12–13 декабря 2019 г. / редкол. : В. А. Прытков [и др.]. – Минск : БГУИР, 2019. – С. 372–373.

2. Ивашенко, В. П. Онтологическая модель пространственно-временных отношений событий и явлений в процессах обработки знаний / В. П. Ивашенко // Вестник БрГТУ. – 2017. – №5(107). – С. 13–17.
3. Ивашенко, В. П. Операции управления массивами данных в линейно адресуемой памяти / В. П. Ивашенко, С. В. Синцов // Доклады БГУИР, №10, – Минск, 2016. – С. 86–93.
4. Ивашенко, В. П. Принципы платформенной независимости и платформенной реализации OSTIS / В. П. Ивашенко, М. М. Татур // OSTIS. 2016. С. 145–150.
5. Ивашенко, В. П. Модели обработки информации в интеллектуальных системах, основанных на семантических технологиях / В. П. Ивашенко, А. С. Бельчиков, А. П. Еремеев // Информационные технологии и системы 2016 (ИТС 2016) : материалы международной научной конференции (БГУИР, Минск, Беларусь, 26 октября 2016) / редкол. : Л. Ю. Шилин [и др.]. — Минск: БГУИР, 2016. — С. 106–107.
6. Ивашенко, В. П. Распределение памяти в неограниченном линейном адресном пространстве / В. П. Ивашенко // Информационные технологии и системы 2019 (ИТС 2019) = Information Technologies and Systems 2019 (ITS 2019) : материалы международной научной конференции, Минск, 30 октября 2019 г. / Белорусский государственный университет информатики и радиоэлектроники; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2019. – С. 110–111.
7. Ivashenko, V. P. String processing model for knowledge-driven systems / V. P. Ivashenko // Doklady BGUIR. – Vol. 18 №6, – Минск, 2020. – pp. 33–40.
8. Ивашенко, В. П. Применение мультиагентного подхода для реализации виртуальной графодинамической среды / В. П. Ивашенко, А. Л. Кондратенко // Четвертая международная летняя школа-семинар по искусственному интеллекту для студентов и аспирантов : сб. науч. тр., Республика. Беларусь, Браславские озера, 29 июня – 6 июля 2000 г. / Белорус. гос. ун-т ; редкол. : О. П. Кузнецов (отв. ред.) [и др.]. – Минск, 2000. – С. 154–163.
9. Ivashenko, V. P. Semantic logging of knowledge processing based on binary generated events / V. P. Ivashenko // Pattern Recognition and Information Processing (PRIP 2019) : Proceedings of the 14th International Conference, Minsk, 21–23 May 2019, “Bestprint”, Belarus. – Minsk, 2019. – P. 172–177.

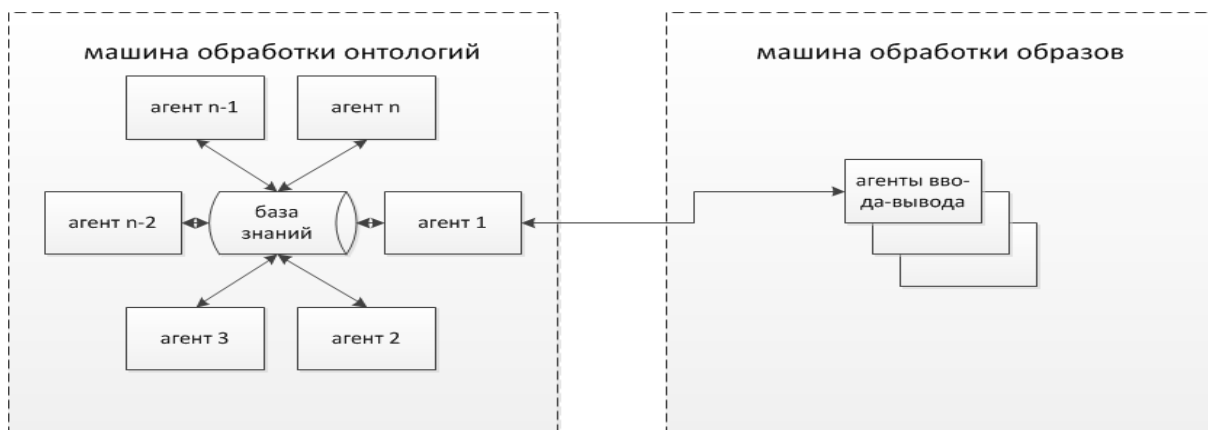


Рис. 1 – Схема взаимодействия компонентов

ПОСТРОЕНИЕ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ИСПОЛЬЗОВАНИЕМ РАДИОЧАСТОТНОГО СПЕКТРА

Карпук А. А., Говорко А. В.
Кафедра программного обеспечения сетей телекоммуникаций,
Белорусская государственная академия связи
Минск, Республика Беларусь
E-mail: A_Karpuk@mail.ru, govorko777@gmail.com

Предложен метод построения онтологии предметной области информационной системы управления использованием радиочастотного спектра. Приведен фрагмент построенной онтологии

ВВЕДЕНИЕ

В работе [1] показано, что в современных информационных системах управления (ИСУ) использованием радиочастотного спектра (РЧС) должны решаться следующие новые задачи:

оценка качества радиосвязи на заявленных номиналах или полосах радиочастот и оценка влияния работы радиоэлектронных средств (РЭС) на заявленных номиналах или полосах радиочастот на работу других РЭС с ранее присвоенными радиочастотами;

выбор заданного количества номиналов или полос радиочастот для работы РЭС с заданными характеристиками и заданным пространственным положением;

оптимизация присвоения номиналов и полос радиочастот для группы РЭС по критерию минимизации взаимных электромагнитных помех.

Для решения новых задач управления использованием РЧС требуются глубокие знания о предметной области сетей радиосвязи. В базе знаний должны содержаться знания о технических характеристиках и координатах РЭС, о технических характеристиках и составе радиолиний и радиосетей, о топографических характеристиках местности (рельефе местности, координатах и типах застройки, координатах лесных массивов и водных поверхностей, электромагнитных свойствах почвы), о метеорологических характеристиках местности (рефракционных свойствах атмосферы, вероятности и интенсивности осадков, тумана, облаков, атмосферных газов и водяных паров) в районе размещения РЭС.

При решении задач должны использоваться знания о трассах распространения радиоволн и влиянии технических характеристик РЭС, топографических и метеорологических характеристик местности на распространение радиоволн. Указанные знания содержатся в технической документации на РЭС, в документации на сети радиосвязи, в электронных картах местности, в документах Международного союза электросвязи.

Для получения формализованного описания предметной области ИСУ использованием

РЧС предлагается построить онтологию предметной области, описанную в настоящей работе.

ОНТОЛОГИЯ ПРЕДМЕТНОЙ ОБЛАСТИ ИСУ ИСПОЛЬЗОВАНИЕМ РЧС

В состав онтологии предметной области ИСУ использованием РЧС предлагается включить множества классов, отношений, функций и аксиом. Поскольку онтология будет использоваться для построения структуры базы знаний ИСУ, то она не будет содержать сведений о конкретных экземплярах классов и отношений.

Множество классов состоит из подмножеств объектов, свойств и доменов. В качестве объектов опишем сущности, предметы, явления, события, которые существуют или могут существовать в предметной области, имеют одно или более свойств, значения которых используются при решении задач в ИСУ использованием РЧС. В качестве свойств опишем возможные свойства (характеристики, атрибуты, реквизиты) объектов. Свойства могут быть однозначными и многозначными. Значением однозначного свойства является элемент домена, с которым связано свойство, вектор из элементов домена или агрегат из значений других однозначных свойств. Значением многозначного свойства является повторяющаяся группа из элементов домена, векторов или агрегатов. В качестве доменов опишем множества возможных значений свойств.

Множество отношений состоит из отношений между объектами и свойствами, между свойствами и доменами, между свойствами, между объектами. Отношения между объектами и свойствами определяют состав свойств каждого объекта. Каждое свойство объекта может быть обязательным или факультативным. Отношения между свойствами и доменами определяют множества допустимых значений каждого свойства. Если множество допустимых значений некоторого свойства зависит от того, у какого объекта имеется это свойство, то такое свойство записывается в онтологию под разными именами столько раз, у скольких объектов оно имеется. Соответственно, для каждого такого свойства в онтологию записывается свой домен. Отноше-

ния между свойствами записываются в онтологию, если имеет место агрегирование свойств (значением некоторого свойства является агрегат из значений нескольких других свойств) или иерархия свойств (значение некоторого свойства зависит от значений других свойств). Отношения между объектами отражают связи «целое-часть», «род-вид», а также иерархические и другие связи между объектами, существующие в предметной области. Большинство отношений онтологии являются бинарными, но допускаются отношения более высокой арности.

Множество функций состоит из n -арных отношений между классами или свойствами, в которых значение элемента с номером n однозначно определяется значениями предыдущих ($n-1$) элементов. С помощью функций описываются первичные и уникальные ключи объектов, иерархические связи между объектами и свойствами, любые другие функциональные зависимости между объектами и свойствами, существующие в предметной области.

Множество аксиом служит для представления в онтологии высказываний об объектах, свойствах, доменах и отношениях, которые всегда истинны. Каждая аксиома формулируется в виде «если <условие на значения доменов для заданных свойств заданных объектов или отношений> то <высказывание о значениях доменов для заданных свойств заданных объектов или отношений>». Аксиомы включаются в онтологию для проверки ограничений на значения свойств, для проверки корректности описания онтологии, для вывода новых истинных высказываний об объектах, свойствах, доменах и отношениях. Сложные аксиомы, определенные на значениях доменов для заданных свойств заданных объектов или отношений, реализуются в виде модулей и библиотек программ, которые могут разрабатываться разными специалистами и добавляться в онтологию.

Приведем состав фрагмента онтологии предметной области ИСУ использованием РЧС, построенной на основе литературы [2, 3]. В подмножество объектов онтологии входят объекты «Полоса радиочастот», «Пользователь РЧС», «Тип передатчика», «Тип приемника», «Тип антенны», «Тип станции наблюдения», «Платформа для размещения РЭС», «Передатчик», «Приемник», «Антенна», «Радиолиния», «Радиосеть», «Частотное выделение», «Частотное присвоение», «Высота местности (матрица высот)», «Область застройки», «Лесной массив», «Водная поверхность», «Область почвы», «Вертикальный градиент коэффициента рефракции атмосферы», «Метеорологические характеристики местности».

В качестве свойств объекта «Тип передатчика» используются свойства «Номенклатура типа передатчика», «Код перестройки частоты», «Единица измерения частот», «Нижняя грани-

ца диапазона частот», «Верхняя граница диапазона частот», «Коды класса излучения», «Количество предварительно установленных каналов», «Единица измерения разноса каналов», «Разнос каналов», «Код перестройки мощности», «Код типа мощности», «Единица измерения мощности», «Нижний предел изменения мощности», «Верхний предел изменения мощности», «Код типа выходного каскада», «Код импульсной модуляции», «Код возможности перестройки длительности импульса», «Единица измерения длительности импульса», «Нижний предел диапазона длительности импульса», «Верхний предел диапазона длительности импульса», «Код возможности перестройки частоты повторения импульсов (PRF)», «Нижний предел PRF», «Верхний предел PRF», «Затухание на второй гармонике», «Затухание на третьей гармонике».

В качестве свойств объекта «Тип приемника» используются свойства «Номенклатура типа приемника», «Код перестройки частоты», «Единица измерения частот», «Нижняя граница диапазона частот», «Верхняя граница диапазона частот», «Коды класса излучения», «Код типа приемника», «Чувствительность приемника», «Количество предварительно установленных каналов», «Единица измерения разноса каналов», «Разнос каналов», «Единица измерения ширины полосы пропускания», «Ширина полосы пропускания на уровне 3 дБ (20 дБ, 40 дБ, 60 дБ)», «Код типа смесителя», «Единица измерения промежуточной частоты», «Промежуточная частота (ПЧ)», «Единица измерения ширины полосы по ПЧ», «Режим гетеродина», «Подавление зеркальной частоты».

Домен «Коды перестройки частоты» состоит из значений «F» (фиксированная частота), «S» (постоянно перестраиваемая ступенями), «T» (постоянно перестраиваемая). Домен «Единица измерения частот» состоит из значений «H» (Гц), «k» (кГц), «M» (МГц), «G» (ГГц). Домен «Коды перестройки мощности» состоит из значений «T» (перестраиваемая мощность), «F» (фиксированная мощность). Домен «Единицы измерения мощности» состоит из значений «U» (микроватт), «L» (милливатт), «W» (ватт), «K» (киловатт), «M» (мегаватт), «G» (гигаватт).

СПИСОК ЛИТЕРАТУРЫ

1. Карпук, А.А. Новые задачи управления использованием радиочастотного спектра и пути их решения / А.А. Карпук // Информационные технологии и системы 2018 (ИТС 2018): материалы международной научной конференции. – Минск: БГУИР, 2018. – С. 112-113.
2. Справочник по компьютерным технологиям управления использованием радиочастотного спектра (КТ). Издание 2015 года. – МСЭ-Р, 2015. – 192 с.
3. Карпук, А.А. Системы оценки качества радиосвязи и оптимизации присвоения частот радиолиниям: монография / А.А. Карпук. – Воронеж: Научная книга, 2015. – 230 с.

ОБОБЩЕННАЯ МОДЕЛЬ МЕДИЦИНСКИХ ЭКСПЕРТНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Курочкин А. В.

Кафедра интеллектуальных систем, Белорусский государственный университет

Минск, Республика Беларусь

E-mail: alex.v.kurochkin@gmail.com

В статье предлагается единый описательный подход к разработке и интеграции медицинских систем для хранения и обработки данных пациентов и обследований, а также к использованию систем поддержки принятия решений и экспертных систем на основе предоставленных данных. Вводится концепция метамодели, используемая для обобщения и упрощения доступа к данным и их проверки независимо от конкретного решения для базы данных, для обеспечения общего API взаимодействия между приложениями и для обеспечения возможности разработки и проверки статистических и прогнозных моделей на основе правил.

ВВЕДЕНИЕ

На сегодняшний день задачи, связанные со сбором и обработкой информации о пациентах медицинского учреждения, включая информацию об историях болезней и результаты диагностических и терапевтических мероприятий, а также задачи, связанные с подбором оптимальных стратегий диагностики и лечения на основании имеющихся данных с использованием медицинских экспертных систем и систем поддержки принятия решений, привлекают все больше внимания. Обработка всех имеющихся данных о пациенте в совокупности упрощает диагностику, поскольку позволяет поставить комплексный диагноз, а интеграция процесса диагностики с медицинскими экспертными системами упрощает саму диагностику и повышает качество оказываемых медицинских услуг.

Одним из препятствий к взаимной интеграции различных медицинских информационных систем в рамках одного учреждения здравоохранения является отсутствие единой модели взаимодействия. В медицинских информационных системах распространены различные виды баз данных, диагностического оборудования и форматов хранения и передачи информации.

В работе рассмотрены основные требования, предъявляемые к медицинским экспертным системам и предложена обобщенная модель медицинской информационной системы, в соответствии с которой любая информационная система может реализовать обобщенный интерфейс для взаимодействия.

1. МЕТАМОДЕЛЬ ДАННЫХ МЕДИЦИНСКИХ СИСТЕМ

В основе любой информационной системы, оперирующей некоторым набором структурированной информации, лежит так называемая модель данных – декларативное описание тех сущностей и их характеристик, которые могут быть выделены в рамках этого набора информации.

Можно выделить две основные проблемы работы с логической схемой данных – сложность проецирования логической схемы на схему какого-либо хранилища или СУБД и вариативность самой схемы.

Для решения этой проблемы и формирования обобщенной структуры для работы с различными видами данных предлагается использовать концепцию метамодели (рисунок 1).

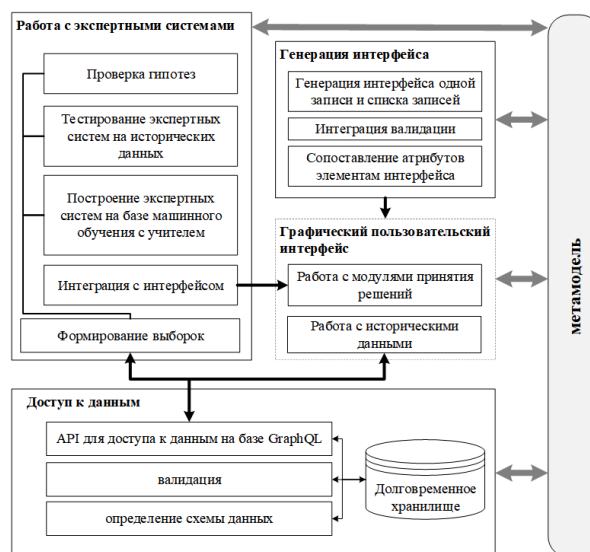


Рис. 1 – Структурная схема основных модулей медицинской информационной системы и их связь с метамоделью

Метамодель представляет собой модель данных, которая используется для описания формата и структуры информации, которой оперирует конкретная медицинская система или хранилище. Метамодель по своим задачам приближена к языкам описания схемы, таким, как DDL (Data Definition Language) в реляционных СУБД на базе SQL, однако, помимо этого, метамодель может использоваться для генерации необходимых представлений для отображения и редактирования данных, а также для их органи-

зации в структуру, пригодную для построения экспертных систем и систем поддержки принятия решений.

Как и язык описания схем в реляционных СУБД, в рамках модели задаётся описание сущностей, атрибутов и их типов, а также их взаимосвязей. Для описания используемых моделей в рамках метамодели предлагается осуществлять на основе существующего открытого стандарта JSON Schema. Описание метамодели с помощью JSON Schema решает две основные задачи – типизация и построение интерфейса доступа к данным, и валидация моделей.

Для большинства информационных систем описанные операции реализуются достаточно однотипно и отличаются только видом и форматом вводимых данных, что вызывает целесообразность обобщения самого механизма генерации пользовательского интерфейса. В общем случае, метамодель содержит достаточно данных для того, чтобы сгенерировать графический интерфейс по описанию полей сущностей, которые используются в рамках конкретной медицинской системы.

II. ИСПОЛЬЗОВАНИЕ МЕТАМОДЕЛИ ДЛЯ РАБОТЫ С ЭКСПЕРТНЫМИ СИСТЕМАМИ И СИСТЕМАМИ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Одной из ключевых задач метамодели является упрощение доступа к данным для формирования и верификации экспертных систем и систем поддержки принятия решений.

При работе с экспертными системами, основанными на данных, по данным метамодели может быть сформирован запрос к самой информационной системе для формирования обучающей выборки. При этом в запросе по историческим данным могут быть перечислены поля, которые требуется использовать в качестве входных признаков, а также указаны целевые поля, которые должны быть определены той или иной моделью при формировании систем на основе методов машинного обучения с учителем.

Помимо предоставления единой точки доступа для формирования выборок, разработанные системы поддержки принятия решений могут в последующем быть интегрированы в сгенерированный пользовательский интерфейс медицинской информационной системы.

III. ЗАКЛЮЧЕНИЕ

В работе представлена обобщенная архитектура медицинской информационной системы на базе метамодели. Показано, что описательное задание и типизация сущностей, которые используются в рамках системы, позволяют оптимизировать разработку такой системы путём автоматизированной генерации методов для валидации

данных, работы с долговременным хранилищем, формирования выборок различной сложности, построения графического интерфейса пользователя и предоставления возможности разрабатывать, тестировать и предоставлять специалистам различные прогностические модели на базе экспертных систем и систем поддержки принятия решений.

На базе предложенного обобщенного подхода реализованы 3 системы для интеграции систем поддержки принятия решений в различных областях медицины – экспертная система для прогнозирования хориальности многоплодных беременностей по результатам пренатальной диагностики на базе ГУ РНПЦ «Мать и дитя», экспертная система для определения дегенеративных оптикнейропатий по результатам оптической когерентной томографии и сканирующей лазерной поляриметрии на базе Городского офтальмологического консультативно-диагностического центра УЗ «3-я городская клиническая больница им. Е. В. Клумова», а также экспертная система прогнозирования исхода тромболитической терапии на базе УЗ «Городская клиническая больница скорой медицинской помощи г. Минска».

1. Collen, M. F. The History of Medical Informatics in the United States (Health Informatics) / M. F. Collen, W. E. Hammond // New York: Springer. – 2015. – 777 p.
2. Pianykh, O. S. Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide / O. S. Pianykh // New York: Springer. – 2009. – 602 p.
3. Elmasri, R. Fundamentals of Database Systems / R. Elmasri, S. Navathe // New York: Pearson. – 2015. – 1280 p.
4. Marrs, T. JSON at Work: Practical Data Integration for the Web / T. Marrs // Cambridge, MA: O'Reilly Media. – 2017. – 376 p.
5. Porcello, E. Learning GraphQL: Declarative Data Fetching for Modern Web Apps / E. Porcello, A. Banks // Cambridge, MA: O'Reilly Media. – 2018. – 198 p.
6. Прибушения, О. В. Оценка плацентации при многоплодной беременности с использованием современных экспертных компьютерных программ / О. В. Прибушения, А. В. Курочкин // Сборник научных трудов «Современные перинатальные медицинские технологии в решении проблем демографической безопасности», 10 вып. – 2017. – с. 106–111.
7. Качан, Т. В. Роль искусственных нейронных сетей в выявлении ранней гибели ганглионарных клеток сетчатки у пациентов с дегенеративными оптикнейропатиями / Т. В. Качан, А. В. Курочкин, Е. А. Головатая, Л. Н. Марченко, А. С. Федулов, А. А. Далидович, О. В. Скрышник, Т. А. Муштина // Офтальмология. Восточная Европа – 2019. –Т. 10, № 21. – с. 445–458.
8. Сенько, К. В. Прогнозирование исхода тромболитической терапии у пациентов с ишемическим инсультом на основе применения нейросетевого анализа / К. В. Сенько, А. С. Федулов, А. В. Курочкин, Е. А. Головатая // Неврология и нейрохирургия. Восточная Европа – 2020. –Т. 10, № 3. – с. 353–366.

СРЕДСТВА ОРГАНИЗАЦИИ И УПРАВЛЕНИЯ КОЛЛЕКТИВНЫМ ПРОЕКТИРОВАНИЕМ БАЗ ЗНАНИЙ

Синельников П. М., Давыденко И. Т.
Кафедра интеллектуальных информационных технологий,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: pavelsinelnikovbel@gmail.com, davydenko@bsuir.by

В работе проведен обзор существующих методов и средств организации и управления коллективным проектированием баз знаний.

ВВЕДЕНИЕ

В настоящее время появляется все большая актуальность в разработке интеллектуальных систем (ИС) различного назначения в связи с усложняющимися задачами, возникающими в различных предметных областях. Одним из ключевых компонентов ИС является база знаний (БЗ), которая хранит все знания о предметной области, в рамках которой разрабатывается ИС. Разработка БЗ занимает до половины времени разработки всей ИС и является трудоемким процессом, требующим применения определенных методов и средств управления разработкой БЗ. Однако, на сегодняшний день ввиду сложности объекта данных технологий, имеющиеся на сегодняшний день средства и модели имеют множество недостатков.

В данной работе будут рассмотрены методы и средства организации и управления коллективным проектированием БЗ для систем, построенных на основе семантических технологий.

I. МЕТОДОЛОГИИ РАЗРАБОТКИ БЗ И ОНТОЛОГИЙ

В большинстве случаев основой БЗ являются онтологии, следовательно, разработка таких БЗ сводится к разработке онтологий и последующему их наполнению. Онтологии широко используются для решения различных задач в таких областях, как инженерия знаний, обработка естественного языка, интеллектуальная интеграция информации, информационный поиск, проектирование и интеграция баз данных/знаний и т.д. Для поддержки разработки онтологий на сегодняшний день было предложено несколько методологий, регламентирующих процесс разработки онтологий, изучаемый в рамках инженерии онтологий. Инженерия онтологий исследует принципы, методы и инструменты для создания и поддержки онтологий [1].

Методологии инженерии онтологий дают набор руководящих принципов и действий по разработке онтологий. Чтобы помочь инженерам по онтологии и экспертам в области построения онтологий, на сегодняшний день было предложено несколько методологий разработки онтологий. Некоторые методологии были предложены

изначально вместе с возникновением направления онтологического инжиниринга, а некоторые возникли в результате опыта и идей, полученных в ходе разработки онтологий для различных проектов.

Для методологий инженерии онтологий, как и для инженерии программного обеспечения (ПО), существует ряд требований, предъявляемых к данным методологиям, которые связаны как со спецификой инженерии знаний, так и с основными трудностями при разработке ПО:

- поддержка коллективной разработки;
- возможность повторного использования компонентов БЗ и онтологий;
- возможность поддержки совместимости разрабатываемых онтологий;
- независимость от ПО, для которого разрабатывается БЗ.

Среди основных методологий, наиболее сформировавшихся на сегодняшний день и ставшими базовыми в области создания онтологий предметных областей, можно выделить следующие: скелетную методологию Ушолда и Кинга, методологию Грюнингера и Фокса (TOVE), METHONTOLOGY, On-ToKnowledge (OTK), CACTUS, DILIGENT, SENSUS и UPON [2,3,5]. Большинство приведенных методологий, не предоставляют достаточной информации о применяемых в них методах и действиях.

II. ОРГАНИЗАЦИЯ ПРОЦЕССА РАЗРАБОТКИ БЗ И ОНТОЛОГИЙ

Процесс разработки описывает какие действия вам необходимо выполнить для разработки онтологии, однако процесс разработки онтологий не определяет порядок выполнения таких действий. Его цель - определить список действий, которые необходимо выполнить. Процесс разработки онтологий в общем случае описывает набор действий [1]:

- планирование основных задач: приоритеты, сроки, ресурсы (исполнители, программное и аппаратное обеспечение);
- определение основных целей, предполагаемого использования, конечных пользователей;

- поиск источников знаний, которые будут использоваться в процессе извлечения знаний;
- получение знаний и создание на основе полученных знаний концептуальной модели предметной области для решения задачи;
- формализация полученной концептуальной модели;
- поиск и интеграция существующих онтологий для расширения разрабатываемой онтологии;
- реализация онтологии на выбранном формальном языке для дальнейшей возможности использования интеллектуальными системами;
- оценка реализованной онтологии с точки зрения корректности и ценности при ее использовании;
- поддержка онтологии (внесение изменений).

Методологии разработки онтологий определяют последовательность и повторяемость данных действий в процессе разработки. Отличным примером такой методологии можно считать METHONTOLOGY [1,5], которая предоставляет достаточно подробную информацию о используемых методах и действиях.

III. СРЕДСТВА РАЗРАБОТКИ БЗ И ОНТОЛОГИЙ

Несмотря на большую распространенность онтологий и БЗ, на рынке программного обеспечения представлено не так много средств для организации разработки таких компонентов.

Исходя из набора действий при разработке любой онтологии, можно выделить основную функциональность, которую должно предоставлять средство поддержки разработки БЗ и онтологий:

- создание новой БЗ;
- редактирование существующей БЗ;
- просмотр БЗ;
- экспорт и импорт в различных форматах;
- документирование БЗ;
- верификация БЗ.

Ввиду того, что над разработкой одной БЗ зачастую трудится команда инженеров, необходимо также предоставлять возможность коллективной разработки. Для этого средства разработки должны предоставить следующие функции:

- изменение БЗ одновременно различными пользователями;
- отслеживание изменений с возможностью их отмены;
- автоматизация проектной деятельности - создание задач, изменение их статусов, приоритетов и исполнителей, обсуждение задач и т.д.;
- использование ролевой модели доступа к фрагментам БЗ.

Наиболее проработанными на текущий момент средствами для разработки онтологий, поддерживающими коллективную разработку являются Collaborative Protege, NeON, CO4 [2]. Все перечисленные проекты имеют свои недостатки, однако можно выделить набор проблем, характерных для большинства средств: отсутствие поддержки ролевой модели, отсутствие автоматизации проектной деятельности/возможности интеграции с системой для управления проектами, отсутствие поддержки полноценного документирования и др.

IV. ИНТЕГРАЦИЯ СРЕДСТВ В КОНТЕКСТЕ РАЗРАБОТКИ БЗ

При создании традиционных систем можно выделить набор средств разработки, которые используют участники проекта: IDE, система для документирования требований, сервер непрерывной интеграции, система управления проектной деятельностью, система контроля версий, системы для code review, анализа проектов, средства коммуникации.

Опыт разработки традиционных систем показывает, что интеграция различных средств, используемых при разработке позволяют повысить скорость разработки а также качество разрабатываемого продукта. Интеграция программных продуктов для работы над проектом в контексте традиционных систем - популярна и востребована, используется в продуктах Atlassian, IntelliJ.

Однако, применение интеграционного подход к разработке средств для создания БЗ и других компонентов ИС является актуальной задачей. В настоящее время время такой подход предложен в рамках Технологии OSTIS [4], предоставляющей комплекс моделей методов и средств разработки интеллектуальных систем, основанных на знаниях.

V. СПИСОК ЛИТЕРАТУРЫ

1. Fernandez, M. METHONTOLOGY: from Ontological Art towards Ontological Engineering / M. Fernandez, A. Gomez-Perez, N. Juristo // Proc. of the AAAI97 Spring Symposium Ser. on Ont. Eng. – 1997. – P. 33–40.
2. Golenkov, V. Principles of organization and automation of the semantic computer systems development / V. Golenkov, D. Shunkevich, I. Davydenko // Open Semantic Technologies for Intelligent Systems (OSTIS-2019). P. 53–90.
3. Iqbal, R. An Analysis of Ontology Engineering Methodologies: A Literature Review / R. Iqbal, A. Murad, A. Mustapha // Sciences, Engineering a. Technology. – 2013. – Vol. 6, № 16. – P. 48–62.
4. IMS metasytem [Электронный ресурс] / IMS metasytem. – OSTIS, 2012. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 25.10.2020.
5. Хала, Е. А. Обзор и анализ методологий и методов построения онтологий / Е. А. Хала // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015) : материалы V междунар. науч.-техн. конф. (Минск, 19-21 февраля 2015 года) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015. – С. 271-274.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ОПТИМИЗАЦИИ ПОДБОРА ПАРАМЕТРОВ ТОРГОВЫХ БОТОВ

Хлопцев А. А., Кузьма Ю. В.

Факультет компьютерных сетей и систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: RedLeonFire@yandex.by, dzotcrew@gmail.com

Задачами данной публикации являются: краткое описание автоматизированной торговли, а также демонстрация того, каким образом можно применить генетический алгоритм к оптимизации подбора параметров торговых ботов. В завершении статьи представлено несколько тестов для сравнения времени работы двух видов генетического алгоритма и алгоритма полного перебора

ВВЕДЕНИЕ

Торговый бот - это программа, реализующая специальный алгоритм, который работает на бирже в режиме реального времени и принимает решения без участия человека. Торговля, при которой совершение сделок происходит с использованием программ-советников (ботов), называется автоматизированной торговлей [1].

Преимущества автоматизированной торговли:

- компенсирование человеческого фактора (влияние эмоций, усталость, и т.д.);
- высокая скорость операций и моментальное реагирование на происходящие события;
- круглосуточное функционирование;
- одновременная работа с большим количеством информации.

Перед тем, как доверить боту деньги и запустить его на настоящем торговом сервере, необходимо провести качественное тестирование программы, чтобы:

- исключить наличие ошибок в боте;
- проверить алгоритм на работоспособность;
- ускорить получение результатов работы алгоритма.

Для тестирования торговых ботов применяются специальные тестировочные модули, которые встроены в торговые терминалы. В торговом терминале MetaTrader компании MetaQuotes Software Corp. данный модуль называется Тестировщик (Tester), в терминале AlgoTerminal компании Soft-FX – Бектестер (Backtester). Названные модули позволяют запустить торговый бот на историческом промежутке данных, на котором эмулируется его работа, что позволяет сравнить результаты торговли при различных наборах входных параметров и выбрать самый оптимальный (прибыльный) из них. Обычно для генерации наборов входных параметров ботов используется алгоритм полного перебора, но с увеличением количества параметров работа данного алгоритма начинает занимать очень много времени, поэтому требуется найти более эффективное решение. Одним из вариантов решения дан-

ной задачи является применение генетического алгоритма.

I. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации и отбор. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе [2].

Поскольку алгоритм самообучающийся, то спектр применения крайне широк:

- задачи на графы;
- задачи компоновки;
- составление расписаний;
- создание «Искусственного интеллекта» [3].

Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора генов, где каждый ген может быть числом или другим объектом.

Некоторым, обычно случайным, образом создаётся начальный набор генов. Он оценивается с использованием «функции приспособленности», в результате чего с каждым набором ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо данный набор решает поставленную задачу.

Из полученного множества с учётом значения «приспособленности» выбираются наборы (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «скрещивание» и «мутация», результатом чего является получение новых наборов. Для них также вычисляется значение приспособленности,

и затем производится отбор («селекция») лучших наборов в следующее поколение.

Эти действия повторяются многократно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма.

Таким образом, можно выделить следующие этапы генетического алгоритма:

- задать целевую функцию (приспособленности) для особей популяции;
- создать начальную популяцию;
- начало эволюционного цикла:
 1. размножение (скрещивание);
 2. мутирование;
 3. вычислить значение целевой функции для всех особей;
 4. формирование нового поколения (селекция);
 5. если выполняются условия остановки, то конец эволюционного цикла [2].

II. ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ НАСТРОЙКИ ТОРГОВОГО БОТА

Для работы боту необходим набор входных параметров, которые применяются в торговом алгоритме. Чтобы ограничить область поиска оптимального набора, для каждого параметра должны выполняться следующие условия:

1. параметр не должен быть больше некоторого значения Max ;
2. параметр не должен быть меньше некоторого значения Min ;
3. разница между параметром и его минимальным значением должна быть кратна некоторому значению $Step$.

Таким образом, каждый параметр должен удовлетворять системе ограничений (1).

$$\begin{cases} value \leq Max \\ value \geq Min \\ value = Min + i \cdot Step, i \in \mathbb{Z} \end{cases} \quad (1)$$

Теперь представим задачу подбора параметров так, чтобы её можно было решить с помощью генетического алгоритма:

- параметр является единичным геном;
- набор входных параметров представляет собой вектор генов;
- функция приспособленности – результат тестирования бота с данным набором параметров (чистая прибыль);
- в качестве мутации выберем функцию изменения текущего значения параметра на

некоторое число, удовлетворяющее системе (1).

Примерное время работы генетического алгоритма в рамках поставленной задачи вычисляется по формуле (2):

$$T = t \cdot countH \cdot (n \cdot \log(n) + n + mut), \quad (2)$$

где

- t – время работы бота на отрезке;
- $countH$ – количество поколений;
- n – количество элементов в поколении;
- mut – количество мутаций в поколении.

III. ТЕСТИРОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Для измерения времени подбора параметров были выбраны 2 разновидности генетического алгоритма и стандартный алгоритм полного перебора. Результаты тестирования представлены в таблице 1.

Таблица 1 – Сравнение времени работы различных алгоритмов

Количество наборов параметров	Полный перебор, мс	Генетический алгоритм с равномерной селекцией, мс	Генетический алгоритм с рулеточной селекцией, мс
5	624	1041	1101
20	1494	1703	1802
100	4163	2117	2351
500	19002	4160	4181
1000	78889	5498	5701
7500	249158	5501	5766
15000	500983	7023	7478

ЗАКЛЮЧЕНИЕ

Из таблицы 1 видно, что если тестируемых наборов параметров не много (меньше 20), то для генерации наборов входных параметров лучше использовать алгоритм полного перебора. Иначе наблюдается сильный рост времени работы данного алгоритма, из чего можно сделать вывод, что при больших наборах параметров выгоднее использовать генетический алгоритм.

1. Технический анализ фьючерсных рынков: Теория и практика / Д. Д. Мэрфи – «Альпина Диджитал», 2011. – 479 с.
2. Алгоритмы эволюционной оптимизации / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2020. – 940 с.
3. Алгоритмы. Руководство по разработке. – 2-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2016. – 720 с.

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПОДДЕРЖКИ ПОДГОТОВКИ И ПРОВЕДЕНИЯ МЕРОПРИЯТИЙ

Черников И. А.

Кафедра интеллектуальных информационных технологий,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: chernik97@gmail.com

Рассматривается онтологическая модель поддержки подготовки и проведения мероприятий, реализованная на базе системы поддержки и подготовки проведения конференций OSTIS.

ВВЕДЕНИЕ

Ежегодно в мире проводятся сотни тысяч деловых мероприятий различного формата: визиты, семинары, конференции, симпозиумы и так далее.

Деловые мероприятия проводятся компаниями-организаторами в целях развития бизнеса и укрепления деловых связей, улучшения взаимодействия, привлечения внимания общественности, а также для обмена знаниями и опытом. Территориально мероприятия могут проходить как в одном, так и в нескольких местах. Сейчас популярен формат видеомероприятий, позволяющих значительно сэкономить средства на организацию.

Существуют определенные виды мероприятий такие, как:

- Конференция;
- Семинар;
- Бизнес-семинар;
- Тренинг;
- Выставка;
- Презентация;
- Прием;
- Круглый стол;
- Митап.

Конференции и семинары – это форматы деловых мероприятий, и их успешность зависит от уровня организации и предварительной подготовки. Умение договориться с известными и интересными спикерами, обеспечение сопутствующими атрибутами участников переговоров, реализация заявленной программы – это показатель профессионализма организации конференции.

В связи со сложностью организации и проведения мероприятий и возможностью автоматизации ее деятельности как информационной поддержки всех категорий участников мероприятий, уведомление и регистрация участников, публикация материалов по результатам проведения, была разработана онтологическая модель поддержки подготовки и проведения мероприятий. Предлагаемая модель разработана с использованием семантических сетей [2] с базовой теоретико-множественной интерпретацией,

где основным способом кодирования является SC-код (Semantic Code) [1].

СТРУКТУРА ОНТОЛОГИЧЕСКОЙ МОДЕЛИ

Рассмотрим структуру онтологической модели поддержки подготовки и проведения мероприятий на примере конференции. Организация взаимодействия участников конференции и организационного комитета является неотъемлемой частью организации любой конференции.

Онтологическая модель должна включать в себя следующую информацию:

- информация об изданных материалах конференции (сборники и программы конференции за предыдущие года);
- информация об участниках, организаторах (программный комитет, рабочая группа, организационный комитет);
- информация о мероприятиях, проводимых в рамках конференции (регистрация, конкурсы, доклады);
- и др.

Рассмотрим понятия описывающее заявку на участие в конференции на примере конференции OSTIS. Данное понятие лежит в следующих предметных областях:

- Предметная область участников конференции;
- Предметная область конференции.

Предметные области включает в себя следующие ключевые отношения:

- Форма участия*;
- Название статьи*;
- Докладчик*;
- Учёная степень*;
- Участие в конкурсах конференции*;
- И др.

Заявка имеет различные формы участия. Если форма участия – очная, то заявка должна рассматриваться как спецификация доклада, который будет озвучен (планируется) на конференции. Если форма участия – заочная, то тогда это спецификация статьи, которая будет отправлена для публикации в сборнике.

Учёная степень и звание рассматриваются как абсолютные понятия, то есть потенциальный участник, оформивший заявку, принадлежит на-

пример, множеству докторов наук и/или множеству кандидатов наук.

Участие в конкурсах конференции рассматривается как принадлежность ко множеству участников студенческого конкурса, участников конкурса молодых ученых и тд. Таким образом персона, подавшая заявку, принадлежит к одному из типов конкурсов конференции конкретного года.

Рассмотрим пример заявки на участие в конференции для участника Петрова Ивана Петровича с темой статьи «Система бухгалтерского учета» на рисунке 1.

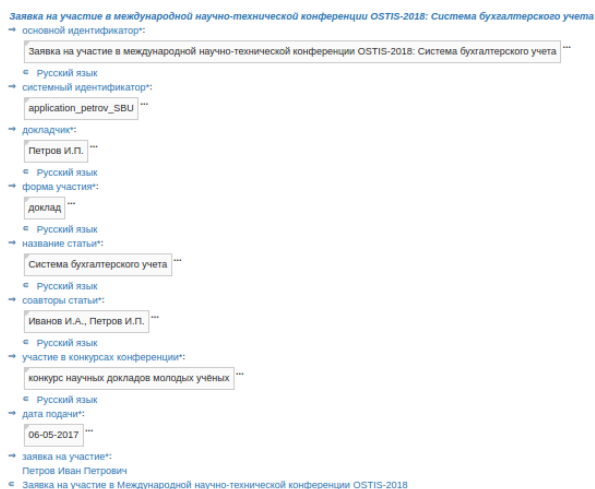


Рис. 1 – Пример заявки на участие в конференции OSTIS-2018 статьи «Система бухгалтерского учета» Петрова Ивана Петровича

Заявка «Система бухгалтерского учета», хранящаяся в базе знаний ИС поддержки и проведения конференции OSTIS, отражает информацию о докладчике (Петров И.А.), соавторах статьи (Иванов И.А. и Петров И.П.), форму участия в конференции (доклад) и форму участия в конкурсах конференции (конкурс научных докладов молодых учёных). Информация о дате заявки (06-05-2017) может использоваться как статистическая и позволит оценить временной промежуток подачи большинства заявок на участие в конференции OSTIS.

Заявка на участие в международной конференции OSTIS-2018 «Система бухгалтерского учета» связана с участником конференции Петровым Иваном Петровичем с помощью отношения – заявка на участие*.

Отношение докладчик* связывает конференцию определенного года с участником конфе-

ренции, который планирует или уже выступил на конференции с докладом.

Каждый для участника конференции отражена информация такая, как электронная почта (petrov@godel.tech.com), ученая степень (не имею), ученое звание (не имею), должность, занимаемая на работе (техник-программист), организация (ИП GodelTech). Город (Минск) и страна (Беларусь) преимущественно относятся к месту работы участника конференции.

Рассмотрим пример участника конференции Петрова Ивана Петровича, представленном на языке SCn (Semantic Code Natural) [1], на рисунке 2.

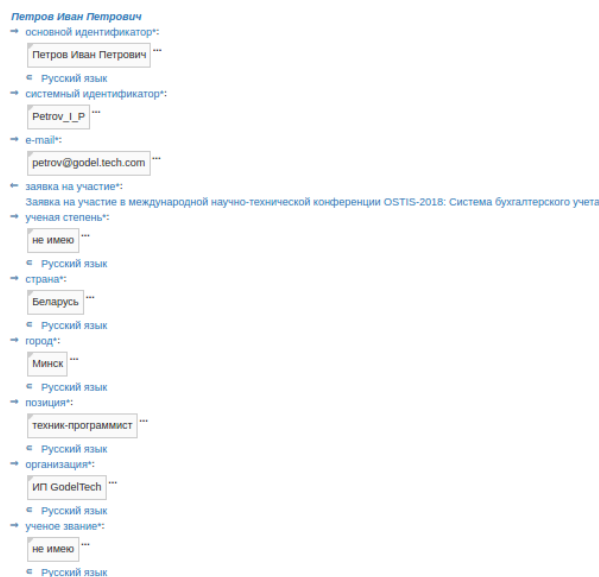


Рис. 2 – Пример участника конференции OSTIS Петрова Ивана Петровича

Выводы

Данная онтологическая модель используется для автоматизации взаимодействия участников и организационного комитета в рамках интеллектуальной системы поддержки и подготовки проведения конференций OSTIS.

СПИСОК ЛИТЕРАТУРЫ

1. Документация. Технология OSTIS. [Электронный ресурс] – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 05.10.2020.
2. Голенков, В.В. Львовский, С.М. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В.В. Голенков, Н.А. Гулякина // Издательство: Минск, 2012. – 35 с.

ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ В ХИМИЧЕСКОЙ ПРОМЫШЛЕННОСТИ

Шангитова Ж. Е., Оспанова Т. Т., Марков А. В.

Кафедра информационных систем, Евразийский Национальный Университет им. Л.Н. Гумилева,
Нур-Султан, Республика Казахстан

Кафедра систем управления, Белорусский государственный университет информатики и
радиоэлектроники, Минск, Республика Беларусь

zhanna.shangitova@mail.ru

E-mail: markov@bsuir.by

В статье исследована актуальность и перспективы применения искусственных нейронных сетей (далее ИНС) в химической промышленности. Рассмотрена классификация и архитектура нейронных сетей и сравнены основные алгоритмы обучения нейронных сетей. Результаты исследования будут использованы в дальнейших статьях.

ВВЕДЕНИЕ

При исследовании каталитических реакций и процессов чаще всего используются методы математического моделирования, которые позволяют описывать изменения состояний изучаемой системы. Однако для некоторых процессов построение полных математических моделей, учитывающих все особенности химико-технологического процесса, является весьма трудоемкой задачей. В этом случае при создании моделей вводят ряд упрощений и ограничений, которые в свою очередь вносят погрешности в численные расчеты и могут существенно исказить получаемые результаты. Альтернативным подходом при моделировании химико-технологических процессов может быть применение ИНС, позволяющих максимально учесть особенности изучаемых процессов. Преимущество ИНС при моделировании химических процессов определяется возможностью строить гибкие модели процессов, которые описываются сложными математическими многомерными функциями. Следует также отметить, что нейронные сети являются наиболее эффективным инструментом при решении задач прогнозирования и управления в условиях, когда имеется связь между переменными-предикторами (входами) и прогнозируемыми переменными (выходами), даже если эта связь имеет очень сложную природу и ее трудно выразить в обычных терминах корреляций или различий между группами. Поэтому создание алгоритмов решения задачи оптимального управления на основе модели ИНС является актуальной задачей [1].

I. ВВЕДЕНИЕ

При исследовании каталитических реакций и процессов чаще всего используются методы математического моделирования, которые позволяют описывать изменения состояний изучаемой системы. Однако для некоторых процессов построение полных математических моделей, учитывающих все особенности химико-

технологического процесса, является весьма трудоемкой задачей. В этом случае при создании моделей вводят ряд упрощений и ограничений, которые в свою очередь вносят погрешности в численные расчеты и могут существенно исказить получаемые результаты. Альтернативным подходом при моделировании химико-технологических процессов может быть применение ИНС, позволяющих максимально учесть особенности изучаемых процессов. Преимущество ИНС при моделировании химических процессов определяется возможностью строить гибкие модели процессов, которые описываются сложными математическими многомерными функциями. Следует также отметить, что нейронные сети являются наиболее эффективным инструментом при решении задач прогнозирования и управления в условиях, когда имеется связь между переменными-предикторами (входами) и прогнозируемыми переменными (выходами), даже если эта связь имеет очень сложную природу и ее трудно выразить в обычных терминах корреляций или различий между группами. Поэтому создание алгоритмов решения задачи оптимального управления на основе модели ИНС является актуальной задачей [1].

II. СТРУКТУРА ИНС

Нейронная сеть – это система, состоящая из многих простых вычислительных элементов, работающих параллельно, функция которых определяется структурой сети, силой взаимосвязанных связей, а вычисления производятся в самих элементах или узлах [2]. Работу искусственного нейрона можно описать показанным на рисунке 1 образом.

III. ЭТАПЫ РАЗРАБОТКИ ИНС

Можно выделить 4 основных этапа в процессе разработки ИНС для решения прикладной задачи [3]:

1. Сбор данных. На этом этапе необходимо составить набор признаков для описания и ясно понимать сущность задачи. После выбора признаков необходимо установить, насколько значимы они и не отражают ли другие существенные признаки.;
2. Преобразование данных. Основная цель преобразования данных заключается в приведении их к одним единицам, например, в интервале $[0;1]$ или $[-1;1]$;
3. Построение модели сети. Одним из важных факторов в построении модели сети является определение ее топологии (рис. 2);

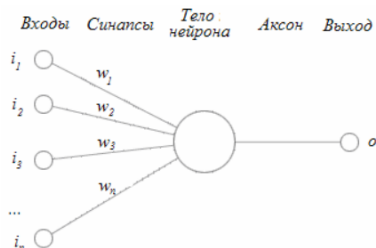


Рис. 1 – Нейрон



Рис. 2 – Топология (архитектура) нейронных сетей

4. Обучение. Немаловажную роль в успешном обучении сети играет активационная функция нейронов. Выделим перечень наиболее используемых функций активации (рис. 3).

IV. АЛГОРИТМЫ ОБУЧЕНИЯ ИНС

Для решения задач обучения могут быть использованы следующие (итерационные) алгоритмы:

- Алгоритмы локальной оптимизации с вычислением частных производных первого порядка.;
- Алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядка;
- Стохастические алгоритмы оптимизации;
- Алгоритмы глобальной оптимизации.

V. ЗАКЛЮЧЕНИЕ

Согласно работе [4] наилучшим образом обучаются сети, методом тренировки которых являются метод градиентного спуска с адаптивным обучением или метод градиентного спуска с учетом моментов и адаптивным обучением. Метод градиентного спуска является наиболее оптимальным методом обучения ИНС и будет использован в дальнейших работах для исследования химического процесса.

VI. СПИСОК ЛИТЕРАТУРЫ

1. Шаймухаметов, Д. Р., Мустафина, С. А., Шаймухаметова, Д. В. Создание искусственной нейронной сети для определения оптимального температурного режима ведения химического процесса // Вест. Башкир.ун-та. – 2019. – № 24. – С. 551–558.
2. Теория нейронных сетей / А. И. Галушкин. – Минск.: ИПРЖР, 2000.
3. Рутковская, Д., Пилиньский, М., Рутковский, Л. Нейронные сети, генетические алгоритмы и нечеткие системы // Горячая линия. – 2006. – С. 452.
4. Федосин, С. А., Ладяев, Д. А., Марьина, О. А. Анализ и сравнение методов обучения нейронных сетей // Вестн. Мордов. ун-та. – 2010. – № 4. – С. 79–88.

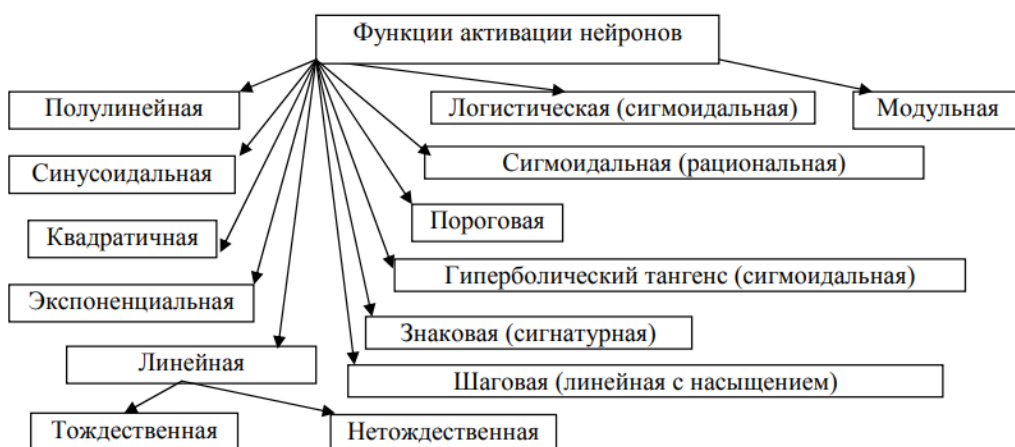


Рис. 3 – Функция активации нейрона

Секция «Проектирование
встраиваемых систем»

Section «Embedded systems design»

ЭКСПЕРИМЕНТАЛЬНОЕ СРАВНЕНИЕ ЭФФЕКТИВНОСТИ РАЗЛОЖЕНИЙ ШЕННОНА И ДАВИО ПРИ СИНТЕЗЕ ЛОГИЧЕСКИХ СХЕМ

Бибило П. Н., Романов В. И.

Объединённый институт проблем информатики Национальной академии наук Беларуси
Минск, Республика Беларусь

E-mail: bibilo@newman.bas-net.by, rom@newman.bas-net.by

На потоке промышленных примеров сравниваются результаты синтеза логических схем в библиотеке проектирования заказных СБИС и FPGA для многоуровневых представлений систем булевых функций. Многоуровневые представления строятся с использованием разложений Шеннона и разложений Давио. Установлено, что использование предварительной логической оптимизации на основе разложений Шеннона позволяет получать более быстродействующие логические схемы меньшей сложности по сравнению с логическими схемами, синтезированными по разложениям Давио.

ВВЕДЕНИЕ

Синтез комбинационных схем в заданном базисе (библиотеке) логических элементов традиционно разбивается на два больших этапа: технологически независимую оптимизацию реализуемых систем булевых функций и технологическое отображение – покрытие оптимизированных представлений описаниями библиотечных логических элементов. Решающее влияние на основные параметры (сложность, быстродействие, энергопотребление) логических схем оказывает первый этап. На данном этапе в качестве одного из основных методов оптимизации в настоящее время является метод на основе разложений Шеннона, когда оптимизируются так называемые бинарные диаграммы решений (англ. Binary Decision Diagram, BDD) [1]. В разложении Давио используются те же подфункции, что и в разложении Шеннона. В данной работе проводится сравнение результатов синтеза логических схем с использованием минимизированных BDD и соответствующих многоуровневых представлений на основе разложения Давио.

1. РАЗЛОЖЕНИЯ ШЕННОНА И ДАВИО

Разложением Шеннона полностью определенной булевой функции $f = f(x)$, $x = (x_1, x_2, \dots, x_n)$ по переменной x_i называется представление

$$f(x) = \bar{x}_i f_0 V x_i f_1.$$

Функции $f_0 = f_0(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f_1(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ называются кофакторами (cofactors) разложения по переменной x_i либо подфункциями. Они получают из функции $f(x)$ подстановкой вместо переменной x_i константы 0 и 1 соответственно. Каждая из подфункций f_0 и f_1 может быть разложена по одной из переменных из множества $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$. Процесс разложения подфункций заканчивается, когда все n переменных будут использованы для разложения, либо когда все подфункции вырождаются до констант 0, 1. На каждом шаге разложения выпол-

няется сравнение на равенство полученных подфункций. Для системы булевых функций разложения Шеннона проводятся по одной и той же последовательности переменных x_i разложения. Разложение Давио использует кофакторы f_0, f_1 и может быть записано в виде *положительного разложения* Давио $f(x) = f_0 \oplus x_i(f_0 \oplus f_1)$, или *отрицательного разложения* Давио $f(x) = f_0 \oplus \bar{x}_i(f_0 \oplus f_1)$. Разложения Давио, аналогично разложениям Шеннона, могут применяться для технологически независимой оптимизации.

II. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Исходными для эксперимента были матричные представления систем $f(x) = (f_1(x), \dots, f_m(x))$ булевых функций [2], по которым с помощью программы BDD_Builder [1] строились BDD-представления, которые переводились в положительные и отрицательные разложения Давио. Все оптимизированные описания представлялись на языке VHDL, являющемся одним из основных входных языков описания проектируемых цифровых систем на элементной базе СБИС. Синтез осуществлялся в системе LeonardoSpectrum с использованием библиотеки [3] проектирования заказных КМОП СБИС и FPGA Virtex 5. Результаты эксперимента приведены в таблицах 1 и 2, соответственно. Площадь S заказных СБИС подсчитывалась в условных единицах, площадь занятых ресурсов FPGA Virtex 5 подсчитывалась в числе LUT (Look-Up Table), задержка t – в наносекундах (нс). Эксперименты показали, что для небольших размерностей систем ДНФ синтезаторы получают одинаковые логические схемы по различным видам разложений. Разложения Давио при прямой схемной реализации в библиотеке проектирования заказных КМОП СБИС имеют более сложные транзисторные схемы, однако промышленные синтезаторы, используя собственные программно реализованные алгоритмы логической оптимизации, не осуществляют локальных преобразований разложений Давио в

функционально эквивалентные и более эффективные разложения Шеннона.

ЗАКЛЮЧЕНИЕ

Использование разложений Шеннона позволяет получать логические схемы с меньшей задержкой и меньшей площади по сравнению с логическими схемами, синтезированными по разложениям Давио. Аналогичные результаты получаются также при реализации тех же примеров систем булевых функций в составе FPGA ф. Xilinx, используя систему проектирования ISE [4].

СПИСОК ЛИТЕРАТУРЫ

1. Бибило, П. Н. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // Программная инженерия. – 2017. – № 8. – С. 369–384.
2. <http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex>. – Дата доступа: 20.03.2018.
3. Бибило, П. Н. Оценка энергопотребления логических КМОП-схем по их переключательной активности / П. Н. Бибило, Н. А. Кириенко // Микроэлектроника. – 2012. – № 1. – С. 65–77.
4. Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов // Издательство: Горячая линия-Телеком, 2020. – 538 стр.

Таблица 1 – Результаты эксперимента 1 в библиотеке проектирования заказных СБИС

Имя примера	n	m	Разложение Шеннона		Положительное разложение Davio		Отрицательное разложение Davio	
			S	t	S	t	S	t
ADD6	12	7	14419	5.97	25395	15.37	25026	13.15
ADDM4	9	8	86959	6.25	159236	13.88	161697	15.73
B12	15	9	16137	2.78	38418	6.91	33686	6.43
BR1	12	8	28112	5.71	80514	13.23	80151	15.58
C880_matr	60	26	30087	6.68	68974	34.95	87344	45.41
Gary	15	11	96389	6.70	259766	23.18	255810	21.34
I7_matr	199	67	21321	1.56	21321	1.51	25339	1.09
IN0	15	11	96389	6.70	259766	23.18	255810	21.34
IN2	19	10	81513	6.47	262154	25.30	250235	25.13
MAX512	9	6	87215	6.60	243009	15.34	239828	14.49
RD73	7	3	16305	3.74	34858	13.21	34942	12.55
RYY6	16	1	3337	2.43	8761	6.87	8348	7.76
TTT2	24	21	49294	4.71	121237	18.15	113603	19.63
X4_matr	94	71	184374	7.40	423678	13.47	441557	18.06

Таблица 2 – результаты эксперимента 2 в библиотеке проектирования FPGA Virtex 5

Имя примера	n	m	Разложение Шеннона		Положительное разложение Davio		Отрицательное разложение Davio	
			LUT	t	LUT	t	LUT	t
ADD6	12	7	23	5	17	5	20	5
ADDM4	9	8	98	6	115	6	138	6
B12	15	9	32	5	35	5	37	5
BR1	12	8	72	5	72	6	71	5
C880_matr	60	26	55	7	68	6	63	7
Gary	15	11	214	6	188	6	214	7
I7_matr	199	67	31	4	31	4	31	4
IN0	15	11	214	6	188	6	214	7
IN2	19	10	225	6	204	6	178	6
MAX512	9	6	132	5	171	6	143	5
RD73	7	3	21	5	31	6	26	5
RYY6	16	1	7	5	5	5	7	5
TTT2	24	21	113	5	95	6	101	6
X4_matr	94	71	333	6	370	6	326	6

ИСПОЛЬЗОВАНИЕ ФЛЕШ-ПАМЯТИ В КАЧЕСТВЕ ИСТОЧНИКА СЛУЧАЙНОСТИ ДЛЯ РЕАЛИЗАЦИИ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

Заливако С. С., Иванюк А. А.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: zalivako@bsuir.by, ivaniuk@bsuir.by

В работе рассмотрены существующие реализации физически неклонлируемых функций (ФНФ) на основе флеш-памяти. Экспериментально показано, что операции чтения без применения кодов коррекции ошибок могут быть использованы при реализации ФНФ. Предложенный подход позволяет избежать использования операций частичной записи, однако требует предварительного сбора статистики о массиве элементов флеш-памяти.

ВВЕДЕНИЕ

В настоящее время физически неклонлируемые функции (ФНФ) используются в качестве альтернативы классической аппаратной криптографии [1]. В силу небольших аппаратных затрат в сравнении с алгоритмами хеширования и шифрования, ФНФ применяются при реализации протоколов аутентификации компактных устройств с низким энергопотреблением (например, устройства Интернета вещей). Аппаратные реализации ФНФ, как правило, основываются на источниках случайности, имеющих физическую природу и, как следствие, уникальность (например, задержка распространения сигнала, частота работы элементов цифровых устройств, пороговые напряжения транзисторов, нестабильность начального состояния элементов памяти и т.п.).

Особенностью организации NAND флеш-памяти является обязательное наличие кодов коррекции ошибок с большой корректирующей способностью (например, код с малой плотностью проверок на четность, LDPC-код). Данная особенность обусловлено крайне низкой надежностью элементов памяти, которые зачастую при повторном чтении могут менять хранимые значения на противоположные. С другой стороны, нестабильные значения бит памяти, а также их позиции являются уникальными и могут быть использованы при реализации ФНФ на основе флеш-памяти [2].

1. СУЩЕСТВУЮЩИЕ РЕАЛИЗАЦИИ ФНФ НА ОСНОВЕ ФЛЕШ-ПАМЯТИ

Реализация ФНФ на основе флеш-памяти, как правило, основана на операции частичного программирования (частичной записи) [3]. Алгоритм извлечения случайности состоит в том, чтобы последовательно записывать стертую страницу и в процессе каждой записи сохранять номера бит, которые меняют свое значение. Операция частичной записи не требует повторного стирания, поскольку интерфейс массива флеш-памяти (Open NAND Flash Interface, ONFI) поддержи-

вает операцию прерывания. Таким образом, ответ ФНФ генерируется как результат непредсказуемого влияния операции частичного программирования на элементы памяти. Недостатком предложенного подхода является использование внутренних команд флеш-памяти, а также в 10-15 раз большая длительность операции записи по сравнению с операцией чтения.

В связи с указанными недостатками предлагается использовать операции чтения без применения кодов коррекции ошибок для извлечения уникальности из массива флеш-памяти. При чтении страницы объемом 4 Кб в порядке 100-150 битах возникают ошибки. Позиции бит, а также частота возникаемых ошибок являются уникальными как для страницы или блока памяти, так и для каждого отдельного идентичного устройства флеш-памяти.

II. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ БЛОКА ФЛЕШ-ПАМЯТИ

Блок – это минимальная единица стирания флеш-памяти. Как правило, современные устройства флеш-памяти содержат от 100 до 1000 страниц в блоке. Данный эксперимент был проведен на плате быстрого прототипирования контроллеров флеш-памяти Cosmos OpenSSD с TLC (Triple Level Cell) памятью производства SK hynix.

Алгоритм эксперимента состоит из 4 шагов: выбор случайного блока, стирание выбранного блока, запись нулей во все страницы блока без использования кодов коррекции ошибок, чтение каждой из страниц по 100 раз. В качестве уникальной характеристики каждой из страниц используется среднее число страниц, полученное при чтении.

Для проведения эксперимента был выбран блок с адресом 0x2F0. Как показано на рисунке 1 число единиц при чтении значительно отличается для различных страниц. Следовательно, аналогично ФНФ на основе кольцевых генераторов, сравнение числа единиц при чтении позволяет генерировать уникальные ответы ФНФ.

Как показано на рисунке 2 на примере страниц 0, 212, 444 и 576 с увеличением числа чтений среднее число единиц на странице стабилизируется. Более того, различия между страницами по среднему соответствуют различиям между страницами при однократном чтении. Следовательно, различия в среднем значении единиц при чтении влияют на стабильность генерируемых бит: чем больше отличие в среднем между страницами, тем стабильнее будет ответ ФНФ.

Следовательно, набор из всех страниц блока является источником случайности для реализации ФНФ.

ЗАКЛЮЧЕНИЕ

Было проведено экспериментальное исследование TLC флеш-памяти производства SK hynix с помощью платы быстрого прототипирования контроллеров Cosmos OpenSSD. Результат эксперимента показал, что количество единиц, полученное при чтении без использования кодов коррекции ошибок, является уникальным

для каждой из страниц в блоке памяти. Предложенный подход позволяет избежать операций частичной записи и генерировать ответы ФНФ с помощью операций чтения. Недостатком предложенного подхода является сбор статистики о среднем числе единиц, полученных при чтении.

СПИСОК ЛИТЕРАТУРЫ

1. Zalivaka, S. S. Design and Implementation of High-Quality Physical Unclonable Functions for Hardware-Oriented Cryptography / S. S. Zalivaka, L. Zhang, V. P. Klybik, A. A. Ivaniuk, C. H. Chang // Secure System Design and Trustable Computing / ed. by C.H. Chang, M. Potkonjak. – New York : Springer, 2016. – P. 39–81.
2. Extracting Robust Keys from NAND Flash Physical Unclonable Functions / J. Shie [et al.] // Proc. of Int. Conf. on Inf. Secur. (ISC'2015), Sep. 2015. – Frankfurt, Germany – P. 437–454.
3. Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints / Y. Wang [et al.] // Proc. IEEE Int. Symp. on Secur. and Priv. (SP'2012), May. 2012. – San Francisco, USA – P. 33–47.

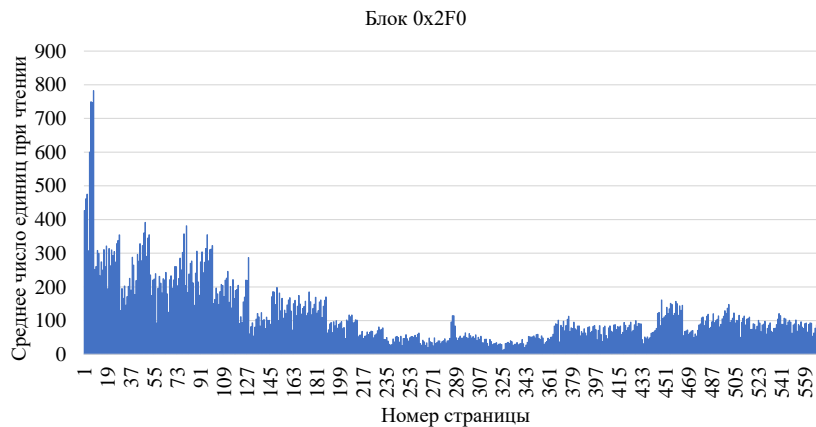


Рис. 1 – Среднее число единиц, полученное при чтении, в зависимости от номера страницы с блока

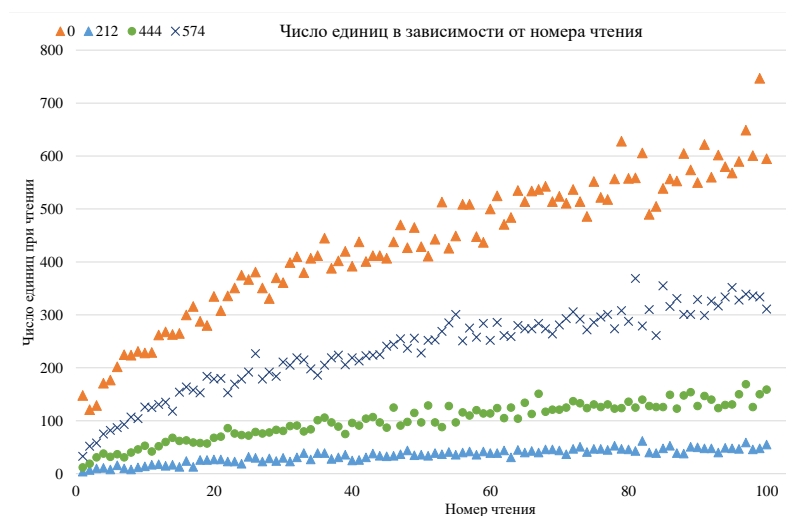


Рис. 2 – Число единиц в зависимости от номера чтения для разных страниц

ОПТИМИЗАЦИЯ ЭНЕРГОЭФФЕКТИВНОСТИ МИКРОКОНТРОЛЛЕРОВ ДЛЯ СХЕМ СО СВЕРХНИЗКИМ ЭНЕРГОПОТРЕБЛЕНИЕМ

Карташов А. А., Журавлёв В. И.

Кафедра систем управления, кафедра теоретических основ электротехники, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: kartashow1998@gmail.com, vadzh@bsuir.by

Рассматриваются основные способы оптимизации энергоэффективности микроконтроллеров для схем со сверхнизким энергопотреблением, их достоинства и недостатки.

ВВЕДЕНИЕ

Приложения со сверхнизким энергопотреблением требуют оптимизации энергопотребления на всех уровнях, от архитектуры микроконтроллера до программного уровня. Оптимальные платформы для разработки эффективных архитектур с одноэлементным питанием могут позволить создавать компактные устройства с наименьшими затратами и максимальной энергоэффективностью [1].

Фактическое потребление энергии в течение всего срока службы устройства – это оценка как среднего, так и пикового энергопотребления. Наиболее эффективными являются следующие методы, используемые в схемах с низким энергопотреблением:

- Эффективное преобразование напряжения;
- Режимы повышения и низкого тока;
- «Умное» управление батареями;
- Дополнительный микроконтроллер.

I. ЭФФЕКТИВНОЕ ПРЕОБРАЗОВАНИЕ НАПРЯЖЕНИЯ

Многие решения со сверхнизким энергопотреблением переходят на архитектуру с одним аккумулятором, чтобы снизить стоимость, размер и вес устройства [2].

Использование источника питания с одним элементом – перспективный вариант, поскольку он упрощает механику держателя батарей и улучшает массогабаритные параметры изделия.

В то время как напряжение от одного элемента батареи обычно находится в диапазоне от 1,2 В до 1,5 В при полной зарядке, элементы могут упасть ниже 1 В, сохраняя при этом значительное количество полезной энергии. Для управления двигателями, светодиодами и даже самим микроконтроллером от одной ячейки требуется регулятор для повышения доступного напряжения до соответствующих уровней.

Современные микроконтроллеры семейства AVR имеют встроенный стабилизатор, который может повышать напряжение до 0,7 В (рис.1), что позволяет продолжать разрядку ближе к исчерпанию резервов элемента.

Интегрированный стабилизатор обеспечивает ток холостого хода 1 мкА (типичный), автоматический запуск при появлении достаточного напряжения и поддержку большинства аккумуляторных технологий. Это позволяет использовать 1,6 В щелочные, 1,5 В литий-ионные, 1,4 В цинковые и 1,2 В NiMH и NiCd, аккумуляторы.

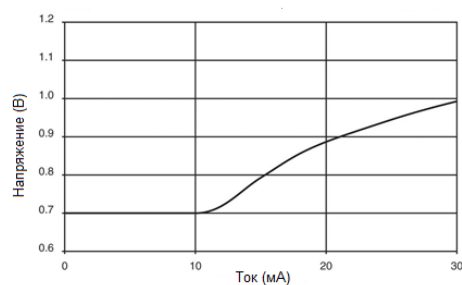


Рис. 1 – Регулирование входного напряжения

II. РЕЖИМЫ ПОВЫШЕНИЯ И НИЗКОГО ТОКА

Высокая токовая нагрузка без схемы внешнего привода также важна для многих приложений. К примеру, повышающий регулятор микроконтроллера ATtiny43U может регулировать ток до 30 мА, что позволяет напрямую управлять светодиодами высокой яркости и небольшими двигателями. На рисунке 2 показана эффективность преобразования ATtiny43U для определенных токов нагрузки в зависимости от оставшегося заряда.

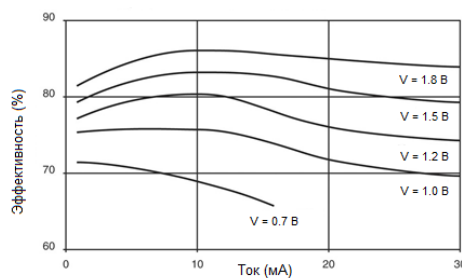


Рис. 2 – Зависимость эффективности токов нагрузки от входного напряжения

Для сохранения эффективности микроконтроллер должен поддерживать несколько режи-

мов работы. Когда устройству требуется высокий ток и строго регулируемое напряжение питания, микроконтроллер и регулятор будут работать в регулируемом режиме[3]. Когда двигатели или другие периферийные устройства не используются, а ток нагрузки падает ниже 0,6 мА, регулятор автоматически переключается в режим низкого тока, выполняя более эффективное преобразование мощности.

Кроме того, при малых нагрузках или при отсутствии нагрузки преобразователь в регулируемом режиме периодически достигает нижнего предела рабочего цикла. При автоматическом переходе в режим низкого тока преобразователь прекращает переключение и снижает потребление тока до минимума, оставаясь при этом активным.

III. «УМНОЕ» УПРАВЛЕНИЕ БАТАРЕЯМИ

Аккумуляторные батареи требуют тщательного мониторинга и контроля заряда в установленных пределах для обеспечения безопасного использования и оптимального срока службы. Аккумуляторы разного химического состава имеют разные пороги напряжения, до которых их можно безопасно заряжать и разряжать.

Предустановленная программа ATtiny43U может контролировать напряжение батареи с помощью встроенного аналого-цифрового преобразователя (АЦП) и решать, когда перейти в режим остановки, таким образом полностью разряжая одноразовые батареи, обеспечивая при этом максимальный срок службы аккумуляторных элементов в течение нескольких циклов перезарядки[4].

Хотя автоматическое отключение микроконтроллера защищает перезаряжаемые элементы, резкое отключение питания может быть неприемлемым с точки зрения приложения. Например, внезапное выключение камеры сделает объектив незащищенным и уязвимым. Использование АЦП для измерения напряжения батареи через регулярные промежутки времени дает приложению возможность привести устройства в безопасную конфигурацию перед выключением.

IV. ДОПОЛНИТЕЛЬНЫЙ МИКРОКОНТРОЛЛЕР

Добавленный в электрическую схему в качестве вторичного процессора микроконтроллер может контролировать такие задачи как:

- обновление дисплея,
- проверка нажатий на клавиатуру,
- «умное» управление батареями.

Использование микроконтроллера таким образом позволяет основному процессору дольше оставаться в спящем режиме без перерывов, что приводит к существенной экономии энергии.

Для микроконтроллеров со сверхнизким энергопотреблением также требуется несколько спящих режимов. Например, программа мониторинга датчика может отслеживать температуру, пока она не превысит пороговое значение. Поддержание работы микроконтроллера в активном режиме во время мониторинга потребляет больше энергии, чем это действительно необходимо. Поддержка различных спящих режимов, которые позволяют отключать различные части устройства, обеспечит лучшее энергосбережение.

Выводы

Микроконтроллеры, объединяющие встроенный регулятор напряжения с настраиваемыми режимами, эффективно устраняют разрыв между минимальным напряжением питания микроконтроллера и выходными напряжениями стандартных технологий одноэлементных батарей, позволяя минимизировать энергопотребление в зависимости от доступных условий нагрузки и напряжения батареи. Имея только одну батарею, отсутствие внешнего регулятора, способность разряжать ячейки до 0,7 В и высокую емкость по току для светодиодов и небольших двигателей, можно создавать компактные, экономичные устройства с батарейным питанием, которые обладают сверхнизким энергопотреблением.

СПИСОК ЛИТЕРАТУРЫ

1. Nikolov H., System-Level Design Methodology for Streaming Multi-Processor Embedded Systems, ISBN 978-90-9024163-0, 2009
2. Optimizing Microcontroller Power Efficiency for Ultra Low Power Designs, Jukka Eskelinen, tinyAVR Marketing Director Kim Meyer, FAE, 2009
3. Andrew Piziali, Functional Verification Coverage Measurement and Analysis, Kluwer Academic Publishers, Norwell, MA, 2004
4. ATtiny43U - Complete Datasheet, Atmel Corporation, 2012

ОРГАНИЗАЦИЯ И ПРОПУСКНАЯ СПОСОБНОСТЬ ИНТЕРФЕЙСА PCI-E УСКОРИТЕЛЬНОЙ ПЛАТЫ ALVEO U250 С УПРАВЛЯЮЩИМ КОМПЬЮТЕРОМ

Качинский М. В., Петровский Н. А., Станкевич А. В.

Кафедра электронных вычислительных средств,

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: {kachinsky, nick.petrovsky, stankevich}@bsuir.by

Рассматриваются вопросы построения интерфейса для обмена данными между Alveo U250 и управляющим компьютером по стандарту PCI Express Gen3. Получена фактическая пропускная способность между ускорительной платой Alveo U250 и хост-компьютером под управлением ОС Linux по шине PCI-e с использованием регистрового AXI4-Lite и отображаемого на память AXI4-MM интерфейсов.

ВВЕДЕНИЕ

Последние несколько десятилетий экспоненциально возрос объем обрабатываемых данных в области машинного обучения, специализированных алгоритмов поиска в базах данных, криптографических системах и т.п. Таким же образом изменился и спрос на высокопроизводительные энергоэффективные вычислительные системы для специализированных задач обработки данных.

Ускорение обработки больших объемов информации только за счет эксплуатации параллелизма процессоров общего назначения и графических карт (CPU/GPU) не всегда эффективно из-за фиксированной архитектуры таких систем [1]. Одним из решений данной проблемы является использование адаптируемых реконфигурируемых ускорителей (ускорительных карт для компьютера) на базе ПЛИС, которые в составе вычислительных систем общего назначения (совместно с CPU/GPU) позволят обеспечить требуемый уровень производительности для решения задач с интенсивными вычислениями.

Фирма Xilinx выпускает семейство адаптируемых карт ускорителей Xilinx Alveo Data Center в форм-факторе PCIe, с широким спектром готовых приложений, которые обеспечивают быстрое развертывание адаптивных ускорителей, что приводит к значительному увеличению вычислительных возможностей и экономии эксплуатационных расходов (TCO — от англ. Total Cost of Ownership) до 65%. Один из таких ускорителей Xilinx Alveo U250 Data Center (далее Alveo U250) представляет собой совместимую с PCI Express Gen3 x16 карту [2]. Карта предназначена для ускорения приложений со сложными вычислительными операциями, таких как машинное обучение, обработка медиаданных.

I. ВЗАИМОСВЯЗЬ ПОДСИСТЕМЫ PCIe И АППАРАТНЫХ ИНТЕРФЕЙСОВ

В настоящем докладе рассматриваются вопросы построения интерфейса для обмена дан-

ными между Alveo U250 и управляющим компьютером.

Alveo U250 выполнена на базе FPGA XCU250, имеющей в своем составе интегрированный блок PCI Express. Для доступа к этому интерфейсу необходимо использовать IP-ядро фирмы Xilinx. В качестве такого ядра может использоваться DMA/Bridge Subsystem for PCI Express v4.1 [3] (далее PCIe-DMA). Данная версия IP-ядра поддерживает, в частности, XCU250.

PCIe-DMA может быть сконфигурирована как высокопроизводительное устройство прямого доступа в память (модуль DMA) для обмена данными между PCI Express и пользовательским устройством, либо как мост между PCI Express и пространством AXI memory. Как модуль DMA, ядро можно использовать для перемещения блочных данных между адресным пространством PCIe и адресным пространством AXI, используя специальный драйвер. При настройке в качестве моста PCIe полученные пакеты PCIe преобразуются в трафик AXI, а принятый трафик AXI преобразуется в трафик PCIe. Функциональность моста идеально подходит для периферийных устройств AXI, которым требуется быстрый и простой способ доступа к PCI Express. В этом режиме ядро может использоваться в качестве конечной точки или корневого порта. Для реализации внутрисистемного интерфейса аппаратной платформы IP-ядро PCIe-DMA целесообразно использовать в режиме DMA, т.к. в этом режиме ядро может быть дополнительно сконфигурировано таким образом, чтобы обеспечить прямое подключение к пользовательской логике (RTL logic). AXI является частью ARM AMBA (Advanced Microcontroller Bus Architecture). В настоящее время используется 4 версия интерфейса.

Существует три типа интерфейсов AXI4 [4]:

- AXI4 memory-mapped (AXI4-MM) — для высокопроизводительных приложений;
- AXI4-Lite — для простых приложений с низкой пропускной способностью (напри-

мер, для чтения/записи регистров управления и состояния);

- AXI4-Stream — для высокоскоростной потоковой передачи данных.

Протоколы с отображением в память AXI4-MM и AXI4-Lite во всех транзакциях используют концепцию целевого адреса в пространстве системной памяти для передаваемых данных. Протокол AXI4-Stream используется для приложений, которые обычно ориентированы на высокоскоростной обмен данными. В этом протоколе понятие адреса не используется. Каждый AXI4-Stream интерфейс действует как один однонаправленный канал, использующий при передаче данных механизм «рукопожатия» (handshake).

PCIe-DMA обеспечивает пересылку данных между памятью хоста и подсистемой прямого доступа к памяти (DMA). Передача данных может быть от хоста к ускорительной карте (H2C) либо в обратном направлении (C2H).

II. АРХИТЕКТУРА СИСТЕМЫ

Простейший вариант подключения PCIe-DMA к пользовательскому проекту заключается в использовании интерфейса AXI4-Lite Master Configuration port. В этом случае доступ хоста к регистрам управления и состояния в пользовательской логике обеспечивается через порт AXI4-Lite Master. Запись и чтение осуществляется 32-разрядными словами. Запрос на чтение или запись от хоста передается через PCIe к AXI4-Lite Master BAR0 (Base Address Register). При чтении данные, передаваемые от пользовательского проекта, возвращаются к хосту через внутреннюю шину CC PCIe-DMA (рисунок 1).

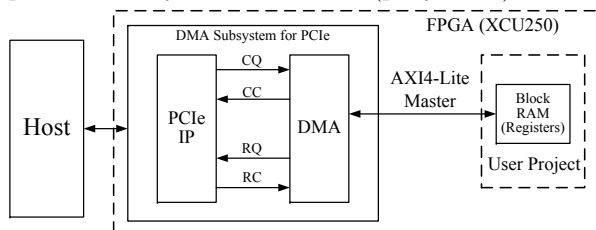


Рис. 1 – Использование интерфейса AXI4-Lite Master

Рассмотренный вариант подключения через интерфейс AXI4-Lite Master обладает относительно невысокой пропускной способностью, однако ее может быть достаточно для определенного класса задач.

Более высокую пропускную способность обеспечивает использование AXI4-MM интерфейса. При этом в пользовательском проекте должна использоваться блочная память с интерфейсом AXI4-MM (рисунок 2). Для H2C передачи PCIe-DMA читает данные от хоста и записывает их в блочную память на пользовательской стороне. Для обратной передачи (C2H) PCIe-DMA читает данные из блочной памяти на пользовательской стороне и записывает их в память хоста.

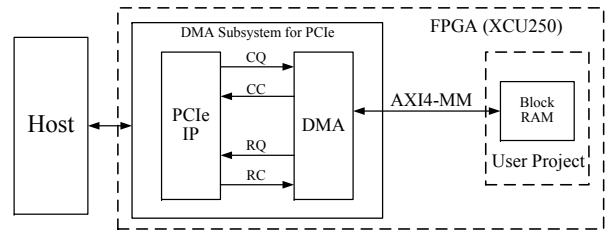


Рис. 2 – Использование AXI4-MM интерфейса

Для реализации дополнительных сигналов управления для доступа к хосту в дополнение к AXI4-MM интерфейсу можно использовать интерфейс AXI4-Lite Master.

Для взаимодействия с IP-ядром PCIe-DMA использовался стандартный драйвер от компании Xilinx — `xdma.ko`. Драйвер обеспечивает высокоскоростной доступ с использованием контроллера DMA к заданными конечными точками устройства (endpoint).

С использованием PCIe-DMA и AXI4-Lite Master реализован регистровый интерфейс. В интерфейсе имеется регистр данных и регистр управления/состояния. Также был реализован интерфейс на основе AXI4-MM.

III. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

При исследовании пропускной способности интерфейсов использовались 2^{20} транзакций (запись/чтение) с последующим усреднением. При исследовании регистрового интерфейса обмен данными производился 32-разрядными словами. Было получено значение пропускной способности 56 Мбит/с. При исследовании интерфейса на основе AXI4-MM изменялся размер пакета (размер транзакции). Наиболее высокая пропускная способность была получена при максимальном размере блока (размере транзакции) равном 32768 байт (размер блока памяти на пользовательской стороне интерфейса AXI4-MM). При указанном размере блока и выключенных прерываниях (`poll_mode = 1`) получена максимальная пропускная способность порядка $\approx 6,8$ Гбит/с при чтении и $\approx 6,7$ Гбит/с при записи.

1. Breathe New Life into Your Data Center with Alveo Adaptable Accelerator Cards. WP499 (v1.0) November 19, 2018; [Электронный ресурс]. – Режим доступа: https://www.xilinx.com/support/documentation/white_papers/wp499-alveo-intro.pdf. Дата доступа: 19.11.2020.
2. ADM-PCIe-7V3; [Электронный ресурс]. – Режим доступа: <https://www.xilinx.com/products/boards-and-kits/1-4i8a6z.html>. Дата доступа: 19.11.2020.
3. DMA/Bridge Subsystem for PCI Express v4.1. Product Guide Vivado Design Suite PG195 (v4.1). June 20, 2019. [Электронный ресурс]. – Режим доступа: https://www.xilinx.com/support/documentation/ip_documentation/xdma/v4_1/pg195-pcie-dma.pdf. Дата доступа: 19.11.2020.
4. AXI Reference Guide UG761 (v14.3). November 15, 2012. [Электронный ресурс]. – Режим доступа: https://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/latest/ug761_axi-reference_guide.pdf. Дата доступа: 19.11.2020.

ОПТИМИЗАЦИЯ ФУНКЦИОНАЛЬНЫХ ОПИСАНИЙ ЛОГИЧЕСКИХ СХЕМ С ИСПОЛЬЗОВАНИЕМ ИНВЕРСНЫХ ПРЕДСТАВЛЕНИЙ

Кириенко Н. А.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

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

Рассматривается новый подход к выполнению оптимизационных преобразований схемы на этапе технологически независимой оптимизации, который заключается в построении и анализе инверсных представлений логических функций. Представлен алгоритм получения инверсных форм. Показано, что рассматриваемое преобразование систем булевых функций позволяет сократить число литералов в функциональном описании.

ВВЕДЕНИЕ

В процессе синтеза логических устройств различают два больших этапа: технологически независимую оптимизацию исходного представления логического устройства и технологическое отображение оптимизированного описания в схему, состоящую из элементов целевой библиотеки проектирования. Этап технологически независимой оптимизации характеризуется выполнением ряда процедур преобразования исходного описания устройства с целью сокращения значений его характеристик – числа промежуточных переменных, конъюнкций, рангов логических выражений и др. Одним из часто используемых критериев качества описания схемы является число литералов, содержащихся в функциональном описании схемы.

В настоящей работе предлагается использование инверсных представлений логических функций в функциональном описании с целью сокращения числа литералов. Приводятся алгоритм такого преобразования и его исследование на группе примеров.

I. ИСПОЛЬЗОВАНИЕ ИНВЕРСНОГО ПРЕДСТАВЛЕНИЯ ЛОГИЧЕСКОЙ ФУНКЦИИ В ФУНКЦИОНАЛЬНОМ ОПИСАНИИ ЛОГИЧЕСКОЙ СХЕМЫ

Исходное функциональное описание логической схемы чаще всего представлено системой логических уравнений, где входные переменные и функции заданы булевыми значениями.

Рассмотрим пример функционального описания, заданного системой логических уравнений, которая содержит три булевы функции y_1, y_2, y_3 .

$$y_1 = \bar{x}_1 * \bar{x}_2 * \bar{x}_3 + \bar{x}_1 * \bar{x}_2 * x_3 + \bar{x}_1 * x_2 * \bar{x}_3 + \bar{x}_1 * x_2 * x_3 + x_1 * \bar{x}_2 * \bar{x}_3 + x_1 * \bar{x}_2 * x_3 + x_1 * x_2 * \bar{x}_3; \quad (1)$$

$$y_2 = y_1 * x_3 + \bar{y}_1 * \bar{x}_3 + y_1 * x_4; \quad (2)$$

$$y_3 = \bar{y}_1 * x_1 + y_1 * \bar{x}_1. \quad (3)$$

Каждое из уравнений можно охарактеризовать числом литералов – общим числом входящих переменных в правой части уравнения. Повторение переменной считается новым входением и увеличивает число литералов. Таким образом, в уравнении (1) – 21 литерал, в уравнении (2) – 6 литералов, в уравнении (3) – 4 литерала. Согласно предлагаемому подходу выполняется инверсия одной из функций, в нашем случае y_1 , при этом инвертируются все входения переменной y_1 в остальных уравнениях системы. Полученное состояние системы уравнений представлено уравнениями (4 – 6).

$$y_1 = x_1 * x_2 * x_3; \quad (4)$$

$$y_2 = \bar{y}_1 * x_3 + y_1 * \bar{x}_3 + \bar{y}_1 * x_4; \quad (5)$$

$$y_3 = y_1 * x_1 + \bar{y}_1 * \bar{x}_1. \quad (6)$$

Число литералов в уравнении (4) сократилось до 3, в уравнениях (5) и (6) не изменилось. Таким образом, достигнут выигрыш 18 литералов благодаря инвертированию функции y_1 .

В настоящей работе рассматривается алгоритм преобразования функционального описания логической схемы путем инвертирования функций, который в дальнейшем может использоваться в процедурах оптимизации.

II. АЛГОРИТМ ПРЕОБРАЗОВАНИЯ ФУНКЦИОНАЛЬНОГО ОПИСАНИЯ ЛОГИЧЕСКОЙ СХЕМЫ ПУТЕМ ИНВЕРТИРОВАНИЯ ФУНКЦИЙ

Одна из задач алгоритма – выполнить операцию отрицания логического выражения по правилам алгебры логики, затем упростить полученное выражение. Эта операция реализована в виде библиотечного модуля ликвидации промежуточных переменных в системе логических уравнений *SeqSdnf*, который имеется во всех экспериментальных системах [1] логического проектирования, разработанных в лаборатории логического проектирования ОИПИ НАН Беларуси.

Модуль *SeqSdnf* работает с матричными представлениями – системами дизъюнктивных нормальных форм (ДНФ) булевых функций. Он

выполняет подстановку подчиненных ДНФ в сжатом виде с выделением каскадов и проверкой на ацикличность [2]. Над логическим выражением выполняются преобразования алгебры логики с целью его упрощения.

Чтобы использовать модуль *SeqSdnf* для получения инверсного представления булевой функции, необходимо выполнить следующие действия:

1. Создать подсистему уравнений (например, (7, 8)), где выходная переменная уравнения для булевой функции заменяется на некоторую временную промежуточную переменную (например, U в уравнении (8)). В подсистему добавляется уравнение, где выходной переменной присваивается инверсия временной промежуточной переменной (7).

$$\begin{aligned}
 y1 &= \bar{U}; & (7) \\
 U &= \bar{x}1 * \bar{x}2 * \bar{x}3 + \bar{x}1 * \bar{x}2 * x3 + & (8) \\
 &+ \bar{x}1 * x2 * \bar{x}3 + \bar{x}1 * x2 * x3 + \\
 &x1 * \bar{x}2 * \bar{x}3 + x1 * \bar{x}2 * x3 + \\
 &+ x1 * x2 * \bar{x}3.
 \end{aligned}$$

2. Перевести подсистему в формат ДНФ и обработать модулем *SeqSdnf*. В результате будет получено уравнение для инверсного представления выходной переменной, в нашем примере это инверсное представление для переменной $y1$ (4).

Алгоритм преобразования системы логических уравнений строит для каждого уравнения инверсное представление. Затем анализирует полученные варианты уравнений в прямой и инверсной формах, и оставляет в системе уравнение с меньшим числом литералов.

Алгоритм реализован в процедуре *INVFUNC* экспериментальной системы логического проектирования FLC [1], разработанной в лаборатории логического проектирования ОИПИ НАН Беларуси.

III. ОРГАНИЗАЦИЯ ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ

Исследование метода выполнялось в системе FLC. В качестве примеров в эксперименте использовались многоуровневые описания комбинационной логики из набора Berkeley PLA test set [3]. Результаты эксперимента представлены в таблице 1. Для каждого примера выполнялась процедура *INVFUNC*. В столбце 2 приведены числа n и m аргументов и функций системы. В столбцах 3, 4, 5 представлены числа литералов (L_s), дизъюнкций (D_s), конъюнкций (K_s) в исходном представлении схемы. В столбцах 6, 7, 8 представлены числа литералов (L_r), дизъюнкций (D_r), конъюнкций (K_r) в результирующем представлении схемы. В столбце 9 представлена величина относительного выигрыша по числу литералов, как отношение числа литералов в исходном представлении к числу литералов в результирующем представлении.

ЗАКЛЮЧЕНИЕ

Анализируя результаты, представленные в таблице 1, можно сделать вывод о том, что путем построения инверсных форм можно добиться значительного выигрыша по числу литералов в описании. Максимальный выигрыш составил 2,96. В 8 случаях из 13 достигается выигрыш по количеству дизъюнкций, во всех случаях достигается выигрыш по количеству конъюнкций в описании.

1. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // Минск : Беларуская навука. – 2011. – 279 с.
2. Торопов, Н. Р. Минимизация систем булевых функций в классе ДНФ / Н. Р. Торопов // Логическое проектирование. – Мн.: Ин-т техн. кибернетики НАН Беларуси. – 1999. – Вып. 4. – С. 4 – 19.
3. Berkeley PLA test set [Electronic resource] / Mode of access: <http://www1.cs.columbia.edu/cs6861/sis>. Date of access: 03.09.2020.

Таблица 1 – Результаты экспериментального исследования эффективности оптимизации

Пример	n, m	L_s	D_s	K_s	L_r	D_r	K_r	Выигрыш лит.
1	2	3	4	5	6	7	8	9
C8	28, 18	365	103	212	275	86	131	1,33
frg1	28, 3	792	116	673	268	41	222	2,96
alu2	10, 4	730	139	532	721	140	521	1,01
ttt2	24, 21	719	203	449	346	119	145	2,08
frg2	143, 139	2855	801	1528	2110	613	925	1,35
gary	15, 11	3810	431	3368	1549	298	1229	2,46
apex6	135, 951	905	242	4248	884	244	400	1,02
x3	135, 99	1816	523	9618	1483	443	658	1,22
term1	34, 10	997	293	557	649	201	299	1,54
9symml	9, 1	277	70	164	254	71	140	1,09
lal	26, 19	258	67	120	224	72	80	1,15
comp	32, 1	200	55	90	160	55	50	1,25
sct	19, 15	236	76	120	177	50	85	1,33

ОЦЕНКА КЛЮЧЕЙ КОМБИНАЦИОННОГО УСТРОЙСТВА ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК

Кохновский С. И., Иванюк А. А.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: stan.ver.i.esk.slav@gmail.com, ivaniuk@bsuir.by

В работе приведена методика построения комбинационного запутывающего устройства и рассмотрена зависимость Хэммингова расстояния между векторами, которые поступают на вход и выход комбинационного устройства, от свойств ключа и входного вектора.

ВВЕДЕНИЕ

Актуальность защиты авторских прав объектов интеллектуальной собственности обусловлена возрастающей конкуренцией на рынке продуктов сферы информационных технологий. В качестве одного из методов нарушения прав интеллектуальной собственности может быть использовано обратное проектирование интегральных схем [1]. Одним из способов повышения сложности обратного проектирования интегральной схемы может выступать внедрение в схему запутывающего устройства.

I. УСТРОЙСТВО ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК

Запутывающим устройством называют цифровое устройство, задающее отображение множества входных сигналов I на множество выходных O в зависимости от подаваемого ключа из множества ключей K . Множества I , O , K – счётные и конечные, причём $|I| = |O| = n$ и $|K| = m$. Мощности множеств связаны соотношением $m = \frac{n}{2}(n-1)$.

На входные линии комбинационной схемы подаются векторы $input = (i_1, i_2, \dots, i_n), input \in I$ и $key = (k_1, k_2, \dots, k_m), key \in K$. На выходные линии схема возвращает вектор $output = (o_1, o_2, \dots, o_n), output \in O$ – перестановку символов входного вектора. Устройство с простейшей конфигурацией (блок), имеющее параметры $n = 2$ и $m = 1$, принимает на вход векторы $input$ и key , а возвращает вектор $output$. Вариантом реализации и описания логики работы блока могут быть два мультиплексора (см. рис. 1).

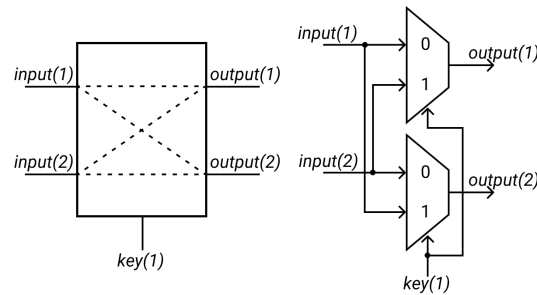


Рис. 1 – Блок

II. МЕТОДИКА ПОСТРОЕНИЯ УСТРОЙСТВА-КОДЕРА

Возможно обобщить схему построения устройства на произвольную размерность (см. рис. 2). Пусть блоки размещаются на линиях $\{1, 2\}, \{2, 3\}, \dots, \{n-1, n\}$, далее происходит размещение блоков на линиях $\{n-2, n-1\}, \{n-3, n-2\}, \dots, \{1, 2\}$, что позволяет обеспечить достижимость всех выходов из входов $input(1)$ и $input(n)$. Под линиями будем понимать отрезок, соединяющий $input(i)$ с $output(i)$. Далее повторяются эти же действия: размещение блоков на линиях $\{2, 3\}, \{3, 4\}, \dots, \{n-1, n\}, \{n-2, n-1\}, \{2, 3\}$, позволяющее передать сигнал с входов $input(2), input(n-1)$ на любой из $n-2$ выходов. Устройство сконфигурировано, когда не осталось входов, для которых не обеспечена достижимость любого выхода. Назовём полученное устройство кодером, переставляющим символы входного вектора в зависимости от поступающего ключа.

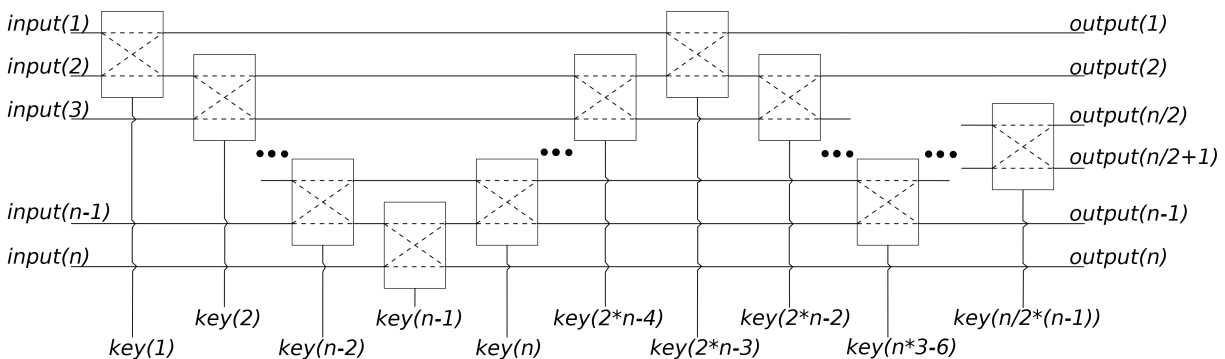


Рис. 2 – Комбинационное устройство размерности $n = k * 2, k \in \mathbb{N}^*$

III. МЕТОДИКА ПОСТРОЕНИЯ УСТРОЙСТВА-ДЕКОДЕРА

Для построения декодера рассматривается устройство размерности $n = 3$ (см. рис. 3). В этом случае при передаче бит ключа в обратном порядке зашифрованная последовательность будет расшифрована устройством-кодером.

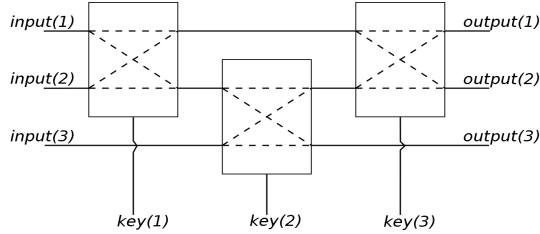


Рис. 3 – Устройство размерности $n = 3$

При изменении логики работы устройства на обратную (подаче бит закодированного вектора на выходы устройства, снятии результата с входов устройства и неизменных конфигурации устройства и ключе), закодированная последовательность будет декодирована, что справедливо для устройств произвольной размерности. Устройство-декодер может быть использовано в качестве кодера для вектора $input$ с произвольным ключом, тогда как устройство-кодер, построенное по вышеописанному алгоритму, будет являться декодером для устройства-декодера на этом же ключе.

IV. ОЦЕНКА ХЭММИНГОВЫХ РАССТОЯНИЙ МЕЖДУ ВЕКТОРАМИ $input$ И $output$

Построим таблицу отображения вектора $input$ на вектор $output$ размерности $n = 3$ в зависимости от вектора key (таблица 1).

Таблица 1 – Всевозможные векторы $output$ для устройства размерности 3

key	$input$							
	000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	000	001	100	101	010	011	110	111
010	000	010	001	011	100	110	101	111
011	000	100	001	101	010	110	011	111
100	000	001	100	101	010	011	110	111
101	000	001	010	011	100	101	110	111
110	000	010	100	110	001	011	101	111
111	000	100	010	110	001	101	011	111

Введём расстояние по Хэммингу $D_H = d(input, output)$ между векторами $input$ и $output$, а также среднее расстояние по Хэммингу $D_H^* = \frac{1}{n} \sum_{i=1}^n d(input_i, output_i)$ в зависимости от ключа key . Замерим D_H в зависимости от подаваемого вектора key , также рассчитаем D_H^* для каждой строки и вычислим среднее расстояние для каждого столбца (таблицы 2, 3).

Таблица 2 – Таблица D_H для устройства размерности 3

key	$input$							
	000	001	010	011	100	101	110	111
000	0	0	0	0	0	0	0	0
001	0	0	2	2	2	2	0	0
010	0	2	2	0	0	2	2	0
011	0	2	2	2	2	2	2	0
100	0	0	2	2	2	2	0	0
101	0	0	0	0	0	0	0	0
110	0	2	2	2	2	2	2	0
111	0	2	0	2	2	0	2	0

Таблица 3 – Таблица D_H^* относительно всевозможных key и средних расстояний для всевозможных $input$

№	000	001	010	011	100	101	110	111
$input$	0	1	1.25	1.25	1.25	1.25	1	0
key	0	1	1	1.5	1	0	1.5	1

Интересными также являются максимальные средние расстояния по Хэммингу для различных размерностей в зависимости от ключа (таблица 4).

Таблица 4 – Таблица максимальных D_H^* для различных размерностей

Размерность	3	4	5	6
D_H^*	1.5	1.88235	2.42424	2.95385

Выводы

В результате анализа таблиц Хэмминговых расстояний различных размерностей получены следующие выводы:

- входные векторы $input$ имеют наибольшее среднее расстояние от выходных векторов $output$ для всевозможных ключей, если соотношение количеств единичных и нулевых бит в векторе $input$ максимально близко к 1;
- ключи, обеспечивающие наибольшее среднее расстояние в зависимости от всевозможных векторов $input$, имеют не менее $n - 1$ единичных бит;
- в отдельную группу могут быть выделены ключи, обеспечивающие работу блока на каждой паре линий нечётное количество раз, что положительно сказывается на среднем Хэмминговом расстоянии для подобных ключей.

СПИСОК ЛИТЕРАТУРЫ

1. Privacy protection of VLSI circuits through high level transformation based obfuscation [Electronic resource] / S. Bhuvaneshwari, A. Hemamalini, A. Anbazhagan. – Global Journal of Pure and Applied Mathematics, 2016. – Mode of access: https://www.researchgate.net/publication/319135767_PRIVACY_PROTECTION_OF_VLSI_CIRCUITS_THROUGH_HIGH_LEVEL_TRANSFORMATION_BASED_OBFUSCATION. – Date of access: 18.10.2020.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ВИРТУАЛИЗАЦИИ ПРИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЯХ НА СИСТЕМАХ С ОБЩЕЙ ПАМЯТЬЮ

Логинова И. П.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: irilog.mail.ru

Кратко рассмотрены современные технологии виртуализации и использование виртуальных сред. Проведен анализ их преимуществ и недостатков. Исследовано применение технологии аппаратной виртуализации к организации параллельных вычислений комбинаторно-логических задач на системах с общей памятью.

ВВЕДЕНИЕ

Современные персональные компьютеры имеют шести-, восьми-, шестнадцати- ядерные. Мощности на одном физическом компьютере позволяют перенести в виртуальную среду операционные среды и системы, необходимые для реализации разных вычислительных задач. Это возможно сделать с помощью современных технологий виртуализации. Основным действующим объектом в любой технологии виртуализации является так называемый гипервизор: Virtual Machine Monitor (VMM) – программная прослойка между гостевыми операционными системами (ОС) и аппаратным обеспечением основной хост-машины, которая позволяет создавать множество гостевых виртуальных машин (ВМ). Каждая ВМ может обладать своей собственной ОС. Виртуализация может быть реализована разными способами, с помощью которых достигаются одинаковые результаты через разные уровни абстракции. У каждого способа есть свои достоинства и недостатки, но каждый из них находит свое место в зависимости от области применения. настоящее время весь рынок виртуализации для платформы x86 можно поделить на классы: 1) *Симуляция процессора.* 2) *Эмуляция аппаратных средств.* 3) *Программная полная виртуализация.* 4) *Программная паравиртуализация* (код, касающийся виртуализации, переносится в гостевую ОС). 5) *Технология виртуализации уровня ОС* (требует внесения изменений в ядро ОС). Этот подход использован в Solaris Containers, Virtuozzo/OpenVZ в ОС Linux, Linux Containers. 6) *Аппаратная или «нативная» виртуализация.*

ПОДДЕРЖКА АППАРАТНОЙ ВИРТУАЛИЗАЦИИ

Аппаратная виртуализация (АВ) представляет собой эмуляцию нескольких виртуальных процессоров для каждой из гостевых ОС при наличии нескольких ядер в физическом процессоре. *Технологии виртуализации Intel и AMD.* Компаниями реализована поддержка АВ в своих моделях процессоров, при помощи расширенного набора инструкций. Эти техники, не об-

ладая прямой совместимостью, выполняют схожие функции, предполагают наличие гипервизора, управляющего немодифицированными гостевыми ОС, предоставляют возможности для разработки платформ виртуализации без эмуляции аппаратуры. Проблемы обеспечения совместимости и стабильности работы АВ были полностью решены только в последних моделях процессоров. Исследованиями возможностей аппаратных техник виртуализации занимаются: AMD, Intel, Dell, Fujitsu Siemens, Hewlett-Packard, IBM, Sun Microsystems, Microsoft и VMware. Виртуализации Intel получила официальное название Intel Virtualization Technology (сокращенно Intel VT). В дополнение к виртуализации вычислений (VT), реализована технология виртуализации ввода-вывода (VT-d). Доработка компанией AMD процессоров получила официальное название AMD Virtualization (сокращенно AMD-V). AMD также реализована технология виртуализации ввода-вывода (IOMMU).

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

Большинство программных платформ виртуализации поддерживают технологии аппаратной виртуализации Intel и AMD. В ОС, поддерживающих платформы паравиртуализации (такие, как Xen или Virtual Iron), аппаратная виртуализация позволяет запускать неизменные гостевые ОС. В табл. 1 перечислены платформы виртуализации и ПО, поддерживающие технологии АВ. Главной причиной отказа от использования виртуализации, является опасения потерь производительности виртуальной вычислительной среды по сравнению с физической средой. В то же время, имеются примеры, в которых эти потери незначительны. Например, виртуальная платформа Parallels выполняет вычисления на виртуальных графических процессорных устройствах (ГПУ) с производительностью, практически равной производительности физических ГПУ, благодаря использованию технологии Intel VT-d, которая позволяет напрямую назначать ВМ выделенные графические процессоры. Несмотря на упомянутые выше достоинства

Таблица 1 – Платформы и ПО с поддержкой аппаратной виртуализации

Платформа виртуализации	Аппаратные технологии	Назначение
Kernel-based Virtual Machine (KVM)	Intel VT, AMD-V	Виртуализация уровня экземпляров ОС под Linux
Microsoft Virtual PC	Intel VT, AMD-V	Настольная платформа виртуализации для хостовых Windows-платформ
Microsoft Virtual Server	Intel VT, AMD-V	Серверная платформа виртуализации для Windows. Версия с поддержкой аппаратной виртуализации
Parallels Workstation	Intel VT, AMD-V	Платформа виртуализации для Windows и Linux хост-машин
VirtualBox	Intel VT, AMD-V	Настольная платформа виртуализации с открытым кодом для Windows, Linux, Mac OS. Поддержка аппаратной виртуализации отключена по умолчанию
Virtual Iron	Intel VT, AMD-V	Платформа, используя аппаратные техники, позволяет запускать 32-битные и 64-битные неизменяемые гостевые ОС без потери производительности
VMware Workstation VMware Server	Intel VT, AMD-V	Для 64-х битных гостевых ОС нужна поддержка Intel VT (также и для VMware ESX Server). В 32-битных гостевых ОС Intel VT отключен по умолчанию
Xen	Intel VT, AMD-V	Позволяет запускать неизменяемые гостевые ОС, используя аппаратные техники виртуализации

виртуализации, необходимо ответить на вопрос, встающий перед реализацией параллельных вычислений: могут ли виртуализированные системы обеспечить производительность, сравнимую с физической? Ответ на этот вопрос дает оценка производительности параллельного выполнения одной из трудоемких задач, применяемых в системах логического синтеза (раздельная минимизация системы полностью определенных булевых функций). Эксперименты показали, что хорошая производительность в виртуальной среде достижима. Проведено сравнение реальной производительности распараллеленной задачи минимизации на хост-машине с производительностью этой же задачи на гостевой ВМ. В качестве платформы виртуализации выбран Oracle VirtualBox [1]. В качестве ОС выбран Windows 7, в хост системе включена поддержка Intel VT (процессор Intel Core i5-3470, CPU 3.2 GHz, в гостевой ОС выделена память – 8 GB). Оценки производительности распараллеливания на системе с общей памятью, проведены для двух приложений, выполненных с использованием разных компиляторов, которые обеспечивают параллелизм по задачам: MS PPL (библиотеки параллельных шаблонов) и Intel CilkPlus. Отмечена совершенно незначительная разница в производительности параллельных вычислений для виртуальной и реальной ОС.

ЗАКЛЮЧЕНИЕ

Виртуализация вводит дополнительную гибкость в среду исполнения распределенных

вычислений и параллельных вычислений, которая может быть использована для решения важных вопросов при разработке и отладке параллельных программ. Кроме того, изначально трудно определить вычислительную нагрузку и временную сложность выполняемого приложения. Виртуализация позволяет решить данный вопрос более эффективно. При наличии свободных вычислительных ресурсов их легко можно добавить нужной ВМ, например, увеличить объем доступной памяти или добавить процессорные ядра, а процедура «живой миграции» позволяет виртуальным ОС и ее приложениям прерывать вычисления, сохраняя состояние, возобновлять вычисления, балансировать нагрузку на доступных ресурсах и оборудовании. Виртуализация может быть использована также для отладки масштабируемых приложений по мере их разработки. Например, отлаживать приложение на небольшом виртуальном кластере, проводить корректное тестирование приложения в точно имитированной среде выполнения. Несмотря на некоторые нерешенные вопросы, преимущества виртуализации неоспоримы. Использование данной технологии открывает новые возможности для высокопроизводительных систем (HPC), принимая во внимание небольшие потери в производительности по сравнению с реальными системами.

1. Oracle VM VirtualBox® User Manual (англ.) [Электронный ресурс] URL: <https://www.virtualbox.org/manual/UserManual.html> (дата обращения: 17.09.2018).

ОПТИМИЗАЦИЯ АППАРАТНО-ПРОГРАММНОГО СРЕДСТВА СШИВКИ ИЗОБРАЖЕНИЙ

Оверченко А. С.

Факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: alex.overchenko.bsuir@gmail.com

Данная статья описывает улучшения времени работы для разрабатываемого аппаратно-программного средства шивки изображений. Целевой платформой разработки является DSP-процессор ADSP-BF609 от Analog Devices, управляемый операционной системой uClinux. Сборка тестового файла проводилась при помощи компилятора bfip-uclinux-gcc версии 4.3.5.

ВВЕДЕНИЕ

В эпоху наступления цифровой эпохи все чаще и чаще возникает вопрос проверки подлинности пользователей. При живом общении основным способом проверки факта, что человек является тем, за кого себя выдает, является верификация его документа, удостоверяющего личность. Такие платежные системы, как WebMoney и Яндекс.Деньги, используют личные встречи своих представителей с проверяемым человеком для проверки документов на их истинность.

I. РАЗРАБАТЫВАЕМОЕ УСТРОЙСТВО

Для ускорения процесса проверки подлинности паспортов разрабатывается отечественный аналог аппаратно-программного устройства шивки изображений с расширенным световым диапазоном. Под расширенным световым диапазоном понимается возможность получения устройством изображений как в видимых спектрах света: красном, синем, зеленом – так и в обычно невидимых: инфракрасном и ультрафиолетовом.

Получение изображения в ультрафиолетовом спектре света позволяет увидеть специально нанесенные пометки, наличие которых является свидетельством подлинности верифицируемого документа. Пример изображения паспорта Республики Беларусь с видимым ультрафиолетовым диапазоном приведен на рисунке 1 [1].



Рис. 1 – Паспорт РБ под ультрафиолетовой лампой

По требованию заказчика в устройстве отсутствуют механизмы, отвечающие за движение, что привело к необходимости добавления

в устройство двух дополнительных камер, на основании данных с которых и производится обнаружение движения сканируемого объекта. Этот факт послужил причиной необходимости синхронизации данных от основного источника сканирования и камер, отвечающих за получение информации о движении. Как следствие, в устройство была добавлена ПЛИС фирмы Xilinx, с помощью которой обеспечивается настройка устройств получения данных и группирование отсканированной информации для дальнейшей ее передачи на основное устройство управления.

В качестве основного обрабатывающего устройства используется DSP-процессор ADSP-BF609, разработанный фирмой Analog Devices. Основное преимущество использования этого рода процессоров заключается в том, что они предназначены для цифровой обработки сигналов (ЦОС), откуда и получили свое название. Данное преимущество обычно реализуется за счет добавления специфического набора команд. Например команда умножения с накоплением (multiply and accumulate, MAC) используется как основная в большинстве алгоритмов ЦОС и обычно реализуется на соответствующих процессорах за один такт [2].

В качестве операционной системы (ОС) на процессоре используется uClinux [3]. Это открытый проект, который является результатом портирования ОС Linux на микроконтроллеры и микропроцессоры, которые не имеют блока управления памятью (memory management unit, MMU). Он позволяет использовать POSIX-совместимые вызовы системных процедур, предоставляет доступ к сетевым интерфейсам и виртуальной файловой системе для взаимодействия с внешней и внутренней памятью устройства. Наличие совместимых с Linux программных интерфейсов позволяет ускорить разработку встраиваемого программного обеспечения как с помощью сборки и запуска тестируемого приложения на персональном компьютере, так и на конечном устройстве без необходимости покупки дорогостоящих отладочных средств для

взаимодействия с процессором без операционной системы.

Таким образом, в нашей работе будет произведен анализ различных улучшений уже реализованных алгоритмов с различными оптимизациями компилятора.

II. ОПТИМИЗАЦИЯ РАБОТЫ КОДА

После разработки конечного устройства и выбора необходимых алгоритмов для получения корректного результата необходимо проводить их оптимизацию для улучшения конкурентных характеристик разработанного устройства. При идентичном качестве выходных данных пользователя начинает волновать такие параметры работы устройства как время получения ответа и время автономной работы.

Для улучшения этих показателей необходимо увеличивать производительность алгоритмов. Сделать это можно различными способами. Мы рассмотрим лишь некоторые из них:

- удаление операций, результаты которых не используются;
- уменьшение количества операций выделения и освобождения памяти;
- ручная реализация функций для блоков фиксированной длины.

Указанные выше улучшения производятся для алгоритма обнаружения движения сканируемого документа. В качестве входных данных использовался один кадр из ПЛИС, который содержит информацию для всех световых каналов. Размер изображения с каждой камеры составлял 32x32 пикселя. Анализ движения проводился на основе данных только с одной камеры.

Для сравнения было взято несколько различных вариантов оптимизаций, поддерживаемых компилятором:

- оптимизация по размеру (O1);
- оптимизация уровня 2 (O2);
- оптимизация уровня 3 (O3).

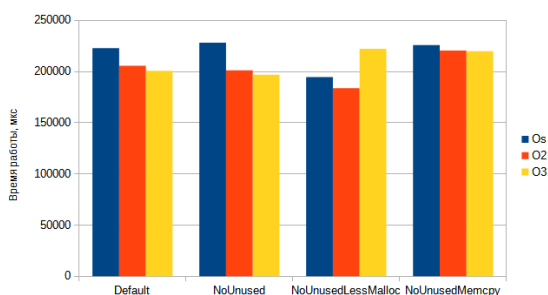


Рис. 2 – Применение ручных оптимизаций для алгоритма обнаружения движения

На рисунке 2 в качестве групп показаны следующие тестируемые варианты (слева направо):

1. изначальный код;
2. удаление неиспользуемых строк кода;
3. удаление неиспользуемых строк кода и уменьшение количество выделений памяти;
4. удаление неиспользуемых строк кода и замена циклов копирования данных на вызовы memsetu библиотеки libc.

Из полученных результатов следует, что различные уровни оптимизации компилятора дают разные значения времени исполнения одиночного обнаружения движения в кадре. В общем случае наблюдается тенденция уменьшения времени работы программы при добавлении оптимизации. Явное использование memsetu вместо циклов показывает увеличение времени работы, что также следует из выдвинутой гипотезы.

Исключение составляет вариант 3 с оптимизацией O3, в котором, предположительно, возникает проблема с кэшем и из-за увеличения размера программы она уже не помещается туда и происходит слишком большое число промахов. Вторым вариантом является то, что адреса объявленных общих массивов не были выровнены, что привело к использованию более медленных команд доступа к памяти или увеличению числа операций обращения к ней.

ЗАКЛЮЧЕНИЕ

В данной статье описаны базовые улучшения кода, которые позволяют получить небольшой прирост в производительности обычно ценой увеличения размера итогового файла. Они подходят для большинства программ, которые готовы пожертвовать одним в сторону другого, но могут не принести желаемого эффекта в приросте производительности.

Поэтому дальнейшим направлением этой статьи является анализ использования таких средств ускорения как OpenMP, специализированных библиотек под DSP-процессоры, обычно написанные производителем, а также различных способов снижения вычислительной сложности алгоритма.

1. Как делают белорусские паспорта и долго ли их печатают? [Электронный ресурс] / Natatnik – Брест, 2017. – Режим доступа: <https://natatnik.by/pasporta/>. – Дата доступа: 15.10.2020.
2. Angoletta, M. E. Digital signal processor fundamentals and system design / M. E. Angoletta – Mode of access: <https://core.ac.uk/download/pdf/44195315.pdf>. – Date of access: 17.10.2020.
3. uClinux – Embedded Linux Microcontroller Project – Mode of access: <http://www.uclinux.org/> – Date of access: 13.11.2018.

ПРОВЕРКА ЭКВИВАЛЕНТНОСТИ СХЕМ НА ТРАНЗИСТОРНОМ УРОВНЕ

Черемисинова Л. Д., Черемисинов Д. И.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: {cld, cher}@newman.bas-net.by

ВВЕДЕНИЕ

Процесс подготовки производства современной СБИС, содержащей более сотни млн. транзисторов, стоит очень дорого: только изготовление набора фотошаблонов требует затрат в несколько миллионов долларов. Поэтому перед изготовлением фотошаблонов обязательно выполняется верификация топологии СБИС (LVS – layout versus schematic verification). Важными инструментами автоматизированного проектирования СБИС, позволяющими снизить трудоемкость верификации топологии электрической схемы, являются средства построения иерархического структурного описания на уровне логических элементов по плоскому структурному описанию на транзисторном уровне. Операция, в результате которой из плоской транзисторной схемы строится иерархическая транзисторная, называется декомпиляцией. Декомпиляция транзисторной схемы является мощным инструментом верификации топологии (позволяя существенно снизить время ее выполнения) [1], а также основой логического перепроектирования (reengineering) интегральных схем [2].

В настоящей работе рассматривается задача верификации транзисторных схем. Предлагается метод и программа проверки эквивалентности пары схем на транзисторном уровне. Входным языком представления транзисторных схем является формат SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [3]. Предложенные средства позволяют, в частности, установить правильность выполнения этапа декомпиляции транзисторной схемы путем сравнения исходной плоской и полученной в результате декомпиляции иерархической транзисторных схем.

I. ПОСТАНОВКА ЗАДАЧИ ПРОВЕРКИ ЭКВИВАЛЕНТНОСТИ ТРАНЗИСТОРНЫХ СХЕМ

Исходными данными декомпиляции является плоская схема из КМОП транзисторов и иерархическая транзисторная схема, в которой блоки второго уровня иерархии соответствуют подсхемам распознанных в результате декомпиляции логических элементов [4]. В схемах, кроме того, допускаются и другие примитивные элементы (биполярные транзисторы, RC-элементы и др.).

Исходная плоская и иерархическая транзисторные схемы представляются в формате про-

ектирования SPICE. В этом формате электрические схемы состоят из элементов, которые соединены друг с другом цепями. Главной частью описания схемы в формате SPICE является список транзисторов, в котором для каждого вывода транзистора (сток, затвор, исток, подложка) указано имя цепи, соединяющей его с остальными частями схемы. Математической моделью представления схемы из транзисторов является помеченный неориентированный двудольный граф. Одну долю графа составляют вершины, соответствующие выводам элементов и портам схемы (входам и выводам всей электрической схемы), а другую – вершины, соответствующие цепям – соединениям между выводами.

Предлагается метод и программа проверки эквивалентности пары схем на транзисторном уровне, которые устанавливают, совпадают ли функционально и топологически транзисторные схемы, и если нет, то указываются различия.

Проверка эквивалентности пары схем на транзисторном уровне может быть выполнена также и инструментами автоматизированного проектирования, решающими задачу сравнения схемы извлеченной из топологии с принципиальной схемой устройства (Logic Versus Schematic check – LVS), например, Mentor Graphics Calibre nmLVS [5], Guardian LVS [6] и другими, при соответствующем преобразовании форматов представления схем. Эти инструменты входят в состав дорогостоящих промышленных САПР проектирования СБИС и для рассматриваемого частного случая транзисторных схем работают значительно медленнее.

II. ПРЕДЛАГАЕМЫЙ ПОДХОД

Подход, положенный в основу этой программы, состоит в том, что задача сравнения транзисторных схем рассматривается как частный случай проблемы изоморфизма графов. В формате SPICE электрические схемы состоят из элементов, которые соединены друг с другом цепями. Формальной моделью описания транзисторной схемы является помеченный неориентированный двудольный граф: одну долю составляют выводы элементов и порты схемы, а другую цепи – соединения между выводами.

Реализованный программой подход к установлению эквивалентности пары транзисторных схем, состоит в сведении этой задачи к проблеме

проверки изоморфизма графов, в которые преобразуются сравниваемые схемы.

Перед сравнением исследуемые транзисторные схемы приводятся, прежде всего, к плоскому виду путем их компиляции (процесса, обратного декомпиляции). Для упрощения задачи проверки изоморфизма графы, представляющие каждую из сравниваемых плоских схем, дополняются ребрами, связывающими все четыре вывода для каждого из транзисторов.

Топология двудольных графов, которые моделируют схемы из транзисторов, имеет некоторые специфические свойства.

1. Исследуемые графы разрежены. В доле цепей двудольного графа, моделирующего схему из транзисторов, подавляющее количество вершин имеют небольшие степени. Имеется только нескольких вершин с большими степенями (например, цепи питания). Таким образом, помеченный граф, моделирующий транзисторную схему, представляет собой разреженный граф.

2. Двудольный граф, моделирующий транзисторную схему, является помеченным: его вершинам могут быть присвоены метки. В схеме транзисторного уровня обычно всего два типа элементов: *n*-МОП и *p*-МОП транзисторы, каждый из которых имеет по четыре вывода, соответствующие стоку, затвору, истоку и подложке. МОП схема моделируется помеченным двудольным графом. Выводы транзисторов (по четыре для МОП-транзистора каждого типа) и порты схемы (входы, выходы схемы) соответствуют вершинам первой доли двудольного графа, а цепи (соединения между выводами) соответствуют вершинам второй доли. Соответственно каждая из вершин первой доли может иметь одну из восьми возможных пометок, если она соответствует выводу транзистора или пометку, больше 8, если соответствует портам схемы.

Необходимым (но не достаточным) условием изоморфизма двух графов является равенство числовых характеристик графов, называемых инвариантами. В качестве таких инвариантов вершин принимаются их степени (полустепени) и число вершин, отстоящих от данной на определенном расстоянии.

Проверка изоморфизма пары графов сводится к построению и сличению их канонических представлений, получаемых путем канонизации графов [7], которая производится путем упорядочения вершин графа в соответствии с инвариантами его вершин, не зависящими от исходной нумерации вершин.

Так как канонизируется помеченный граф, то задача канонизации сводится к изменению пометки (перемаркировке) его вершин. Вначале множество вершин графа разбивается на блоки таким образом, что каждый блок разбиения

содержит вершины с одинаковыми пометками и степенями.

Блоки разбиения упорядочиваются по возрастанию меток входящих в него вершин, а при равенстве меток вершин по возрастанию их степеней. Каждой вершине графа приписывается вектор, *i*-я компонента которого равна числу вершин *i*-го блока, смежных с данной вершиной. Если некоторый блок разбиения содержит вершины с различными векторами, то он разделяется на более мелкие блоки так, чтобы каждый из них содержал вершины с одинаковыми векторами. Вновь полученные блоки заменяют расщепляемый блок и упорядочиваются по возрастанию векторов, соответствующим образом пересчитываются векторы, приписанные вершинам графа.

Процесс деления и канонизации продолжается до тех пор, пока в каждом блоке будут вершины только с одинаковыми векторами.

Проблема канонизации графа в вычислительном отношении так же трудна, как и проблема изоморфизма графов. Однако учет введенных выше особенностей графов, представляющих транзисторные схемы, позволил разработать программу, которая производит проверку идентичности пары транзисторных схем за линейное время от размеров их описаний.

ЗАКЛЮЧЕНИЕ

Разработанная программа верификации была протестирована на практических схемах, содержащих до 500 тысяч транзисторов. Для таких пар схем время выполнения не превышало одной-двух минуты на компьютере с четырехядерным процессором Intel i5-4460 3.20GHz и оперативной памятью 16,0 ГБ.

СПИСОК ЛИТЕРАТУРЫ

1. Nian, Zhang. The subcircuit extraction problem / Zhang Nian, D.C. Wunsch, F. Harary // IEEE Potentials. – 2003. – Vol. 22. – № 3. – P. 22–25.
2. Hunt, V.D. Reengineering: Leveraging the Power of Integrated Product Development / V.D. Hunt. – Wiley. – 1993. – 283 p.
3. Baker R.J. CMOS Circuit Design, Layout, and Simulation (Third Edition). – Wiley-IEEE Press, 2010. – 1214 p.
4. Черемисинов Д.И. Извлечение сети логических элементов из КМОП схемы транзисторного уровня / Д.И. Черемисинов, Л.Д. Черемисинова // Микроэлектроника. – 2019. – 48, № 3 (48) . – С. 224–234.
5. Calibre nmLVS over – Mentor. In view <https://www.mentor.com/products/fv/multimedia/overview/calibre-nmlvs-overview-07b8e2e5-4df9-4384-a8cf-11cdebc14b00>
6. Guardian LVS verification tool – Silvaco. In <https://silvaco.com/examples/expert/section4/example6/index.html>.
7. Закревский, А.Д. Логические основы проектирования дискретных устройств / Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. – М.: Физматлит. – 2007. – 589 с.

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК ФИЗИЧЕСКИ НЕКЛОНИРУЕМОЙ ФУНКЦИИ ТИПА АРБИТР НА ПЛАТАХ БЫСТРОГО ПРОТОТИПИРОВАНИЯ

Шамына А. Ю., Иванюк А. А.

Кафедра программного обеспечения информационных технологий, кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: shamyna@bsuir.by, ivaniuk@bsuir.by

В настоящей работе проведено исследование характеристик физически неклоняемых функций типа арбитр, реализованных на ПЛИС. Описана реализация экспериментальной установки и технология сбора данных. Проведено сравнение характеристик с различными длинами симметричных путей

ВВЕДЕНИЕ

Физически неклоняемые функции (ФНФ) широко используются для защиты цифровых устройств от нелегального копирования, а также в качестве примитивов для генерации случайных числовых последовательностей. Особый интерес для исследования представляют ФНФ типа арбитр (АФНФ)[1], реализованные на FPGA, благодаря гибкости конфигурации и небольшим аппаратным затратам. Стабильность, случайность и уникальность являются важнейшими характеристиками ФНФ. Значения этих характеристик выступают в качестве критериев возможности использования ФНФ в определенных практических применениях. В данной работе проведено исследование характеристик АФНФ на нескольких платах быстрого прототипирования Digilent Nexys-4 с FPGA Artix-7[2].

I. ПОДГОТОВКА ЭКСПЕРИМЕНТА И СБОР ДАННЫХ

Реализация АФНФ предполагает наличие нескольких компонентов: генератор тестовых импульсов, блок симметричных путей (БСП) и схему арбитра. Схематично АФНФ представлена на рис. 1.

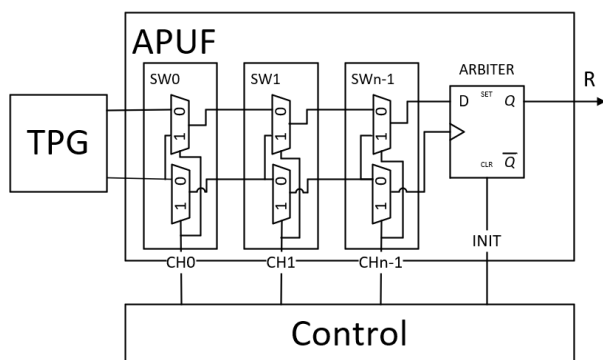


Рис. 1 – АФНФ типа арбитр

Проектное описание АФНФ было создано на языке VHDL с использованием САПР Vivado 2018.2. Передача данных между ПК и ПЛИС

реализована на основе использования стандартных IP-компонент в среде Vivado. Также для проведения экспериментов и передачи данных через интерфейс UART на FPGA был развернут софт-процессор Microblaze[3], что позволило реализовать передачу данных между ПК и ПЛИС более эффективно. Программирование софт-процессора осуществлялось при помощи средств Xilinx SDK. Контроллер АФНФ был создан на основе цифрового конечного автомата (ЦКА).

Т.к. основной целью проведения эксперимента было исследование зависимости характеристик АФНФ от длины симметричных путей, было принято решение расположить арбитры через каждые 8 звеньев БСП. Благодаря этому стало возможным за один эксперимент получить результаты для АФНФ с различными длинами БСП. В данной реализации было выбрано число звеньев БСП $N = 128$. В качестве арбитра использован D-триггер.

Для генерации слабокоррелированных запросов использовался LFSR с внешними сумматорами по модулю два и характеристическим полиномом $\Phi(X) = x^{128} + x^{28} + x^{26} + x^2 + 1$. LFSR был реализован аппаратно, т.к. генерация запросов на стороне ПК и последующая их парадочка на ПЛИС может значительно увеличить время эксперимента.

Для вычисления характеристик АФНФ было проведено $E = 10$ экспериментов на $M = 2$ различных кристаллах. В рамках каждого эксперимента было сгенерировано $C = 10^6$ псевдослучайных запросов. Для чтения данных из COM-порта на ПК и их записи в файлы использовалось ПО Tera Term. Также для анализа полученных результатов было реализовано собственное программное средство на языке программирования C#, включающее в себя функции по анализу стабильности и случайности ответов ФНФ.

II. ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК АФНФ

Стабильность является одной из ключевых характеристик ФНФ. Для классической реализации АФНФ свойственны метастабильные от-

веты на определенные запросы, которые возникают из-за перехода схемы арбитра в метастабильное состояние при минимальной разнице между фронтами тестовых сигналов. Это может негативно сказаться на стабильности ответов и требовать применения дополнительных решений. Для экспериментальной оценки стабильности был использован подход, описанный в работе [4].

Результаты по измерению средней стабильности S_{avg} для АФНФ различной разрядности, полученные в результате эксперимента, показаны на рисунке 2.

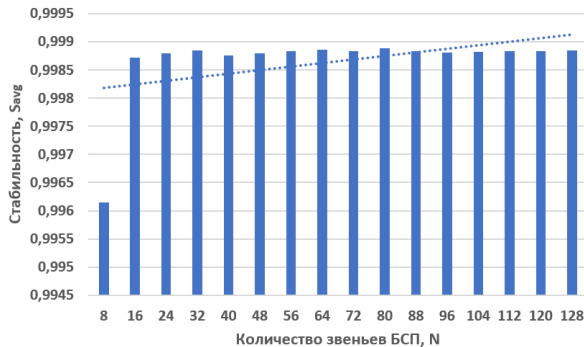


Рис. 2 – Зависимость значения средней стабильности АФНФ от количества пар мультиплексоров

Минимальное значение стабильности составляет $S_{min} = 0.5$ для каждой из реализаций. Доли метастабильных ответов от общего числа представлены на рисунке 3. Полученные результаты демонстрируют увеличение стабильности с увеличением длины симметричных путей.

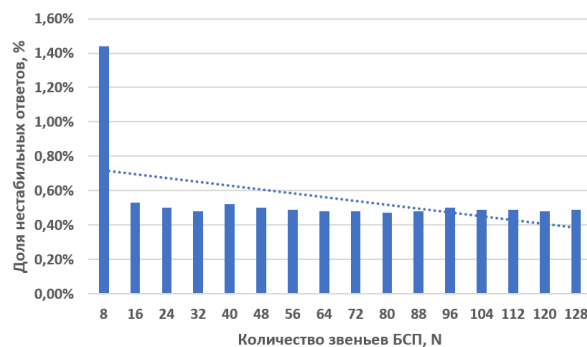


Рис. 3 – Зависимости доли метастабильных ответов АФНФ от количества пар мультиплексоров

Для оценки случайности была взята вероятность появления ответа $r = 1$. Пусть ФНФ сгенерировала последовательность ответов R длиной n , тогда для оценки вероятности появления символа α p_α , встретившегося в R ровно k_α раз является отношение:

$$p_\alpha = \frac{k_\alpha}{n}.$$

Таким образом, рассчитанные результаты p_1 отображены в виде гистограммы на рисунке 4.

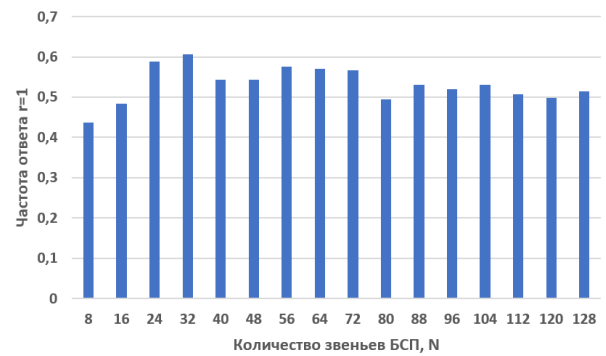


Рис. 4 – Зависимость значения вероятности ответа АФНФ $r = 1$ от количества пар мультиплексоров

Полученные результаты свидетельствуют о стабилизации значения p_1 к эталонному $p_1 = 0.5$ при увеличении количества звеньев БСП.

III. ЗАКЛЮЧЕНИЕ

В ходе экспериментального исследования АФНФ различной размерности были оценены такие характеристики ФНФ, как стабильность и случайность. На основе полученных данных можно сделать вывод об улучшении характеристик ФНФ с увеличением длины БСП. Однако с увеличением длины БСП возрастают аппаратные затраты и время отклика, что может являться существенным ограничением в определенных случаях.

Также при анализе результатов были обнаружены нестабильные ответы для определенных запросов. Это может затруднить использование классической реализации АФНФ в случаях, где требуется высокая стабильность.

В дальнейших исследованиях планируется изучить подходы по увеличению стабильности АФНФ, а также влияние динамических эффектов на характеристики ФНФ.

СПИСОК ЛИТЕРАТУРЫ

1. A technique to build a secret key in integrated circuits for identification and authentication applications / J.W. Lee [et al.] // Intern. Symp. VLSI Circuits (VLSI'04), Honolulu, USA, June 15–19, 2004. – Honolulu, 2004. – P. 176–179
2. Nexys 4 artix-7 FPGA: Trainer board recommended for ece curriculum [Electronic resource]. – Mode of access: <https://store.digilentinc.com/nexys-4-artix-7-fpga-trainer-board-limited-time-see-nexys4-ddr/>. – Digilent, Inc, 2020. – Date of access: 30.10.2020.
3. Microblaze soft processor core [Electronic resource]. – Mode of access: <https://www.xilinx.com/products/design-tools/microblaze.html>. – Xilinx, Inc, 2020. – Date of access: 30.10.2020.
4. Метод увеличения стабильности физически неклонированной функции типа арбитра. / Заливако С.С., Иванов А.А., В.П. Клыбик // Информатика – 2017. – №1(53) С. 31-43.

ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ БЕСКОНТАКТНЫХ СМАРТ-КАРТ ТИПА MIFARE CLASSIC

Шинкевич Н. Н.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: nn5h@yahoo.com

В статье рассмотрены основные уязвимости карт MIFARE Classic, а также реализованы атаки, эксплуатирующие описанные уязвимости. Поточковый шифр Crypto1, который используется для защиты данных на карте, был успешно взломан [2][3], что позволяет в теории восстановить секретные ключи за короткое время. Помимо уязвимостей потокового шифра, будут рассмотрены уязвимости стека протоколов, протестированы атаки. Наибольший интерес представляют те, которые восстанавливают ключ за несколько секунд.

ВВЕДЕНИЕ

Целью настоящей статьи является анализ безопасности смарт-карт MIFARE Classic и стойкости проприетарного шифра Crypto1, проверка работоспособности существующих эксплоитов к известным уязвимостям.

I. ТИПЫ КАРТ MIFARE CLASSIC

Семейство MIFARE Classic состоит из карт 1K-4K, EV1 1K-4K, ID и Mini[6]. Все карты Classic используют потоковый шифр Crypto1 для защиты данных; различаются только размером EEPROM и организацией памяти.

II. ОРГАНИЗАЦИЯ ПАМЯТИ

EEPROM MIFARE Classic организована в виде секторов, разделенных на блоки. В одном секторе обычно содержится 4 блока, 1 блок содержит 16 байт. В 1 блоке данных ($block_0$) 1 сектора ($sector_0$) хранятся данные о производителе чипа и UID; имеет защиту от перезаписи. 4 блок данных ($block_3$) 1 сектора ($sector_0$), «трейлер» [1], хранит ключи A и B и условия доступа.

III. CRYPTO1

Crypto1 — проприетарный алгоритм шифрования (рис.1), созданный NXP. Исследования [2][3], показали, что его безопасность является невысокой.

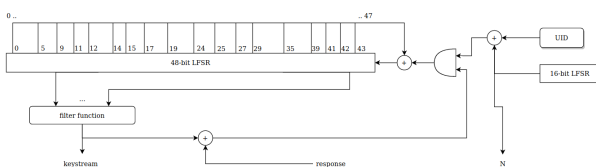


Рис. 1 – Схема шифра Crypto1

48-bit LFSR Начальное состояние определяется секретным ключом а, каждый новый бит keystream генерируется на основании 18 бит состояния РСЛОС в определенный момент времени (рис.1)[2]. **Двухуровневая нелинейная функция** или фильтр-функция (рис.2)[5].

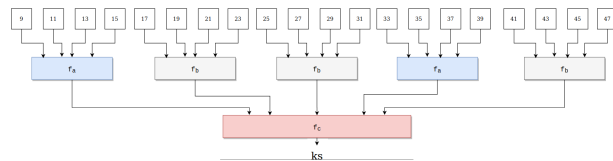


Рис. 2 – Двухуровневая нелинейная функция

16-bit LFSR 16-битный РСЛОС используется картой как ГПСЧ. Генерируемые значения должны быть 32-х битными, это необходимо для корректной работы шифра.

$$L(x_0x_1..x_{15}) = x_0 \oplus x_2 \oplus x_3 \oplus x_5 \quad (1)$$

$$suc(x_0x_1..x_{31}) = x_1x_2..x_{31}L(x_{16}x_{17}..x_{31}) \quad (2)$$

Состояние РСЛОС определяется по ф.1; ($State_i$) генерируются по ф.2[4].

$$suc^n(State_i) = suc(suc^{n-1}(State_i)) \quad (3)$$

$$State_i = x_0x_1..x_{31_i} \quad (4)$$

Для вычисления N_T, N_R, A_T, A_R используется suc^n ф.3-4[4]. **Процедура аутентификации** Для любой операции с данными считыватель должен пройти процедуру аутентификации (рис.3)[5].

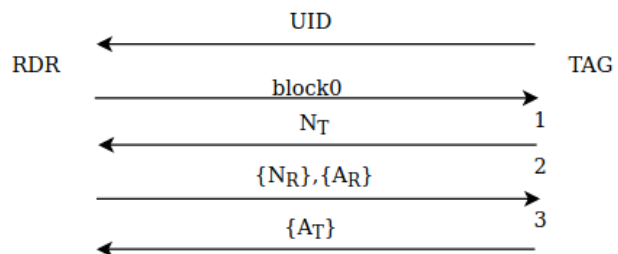


Рис. 3 – Аутентификация

На этапах 1, 2 и 3 показан обмен значениями $N_T, \{N_R\}, \{A_T\}, \{A_R\}$ ф.5-7[4].

$$N_R = suc^{32}(N_T); \{N_R\} = N_R \oplus ks_1 \quad (5)$$

$$A_R = suc^{64}(N_T); \{A_R\} = A_R \oplus ks_2 \quad (6)$$

$$A_T = suc^{96}(N_T); \{A_T\} = A_T \oplus ks_3 \quad (7)$$

IV. Уязвимости

Ненадежный ГПСЧ Карта использует 16-битный РСЛОС для генерации 32-битных значений, откуда имеем $2^{16} - 1 = 65535$ возможных значений (ф.8).

$$n_k \oplus n_{k+2} \oplus n_{k+3} \oplus n_{k+5} \oplus n_{k+16} = 0, k \in [0..15] \quad (8)$$

Неиспользуемые фильтр-функцией биты Биты 0-8 не используются (рис.2), что позволяет реализовать функцию отката состояния регистра до первоначального состояния (ф.9-10).

$$R(x_1 \dots x_{48}) = x_5 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{14} \\ \oplus x_{15} \oplus x_{17} \oplus x_{19} \oplus x_{24} \oplus x_{25} \oplus x_{27} \oplus x_{29} \\ \oplus x_{35} \oplus x_{39} \oplus x_{41} \oplus x_{42} \oplus x_{43} \oplus x_{48} \quad (9)$$

$$R(x_1 x_2 \dots x_{48}) = R(x_1 x_2 \dots L(x_0 x_1 \dots x_{47})) = x_0 \quad (10)$$

Утечка битов ключа через Parity bits Уязвимость позволяет вычислить 3 бита ключа при помощи битов четности, т.к 1-й бит следующего байта шифруется тем же битом ключевого потока, что и бит четности (рис.4).

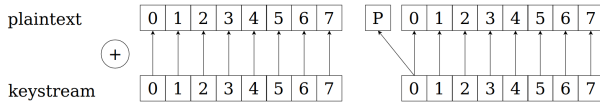


Рис. 4 – Вычисление битов четности

Утечка битов ключа через код ошибки

При условии правильности parity bits, и неверности ответа, код HALT (0x5) будет зашифрован 4-мя битами ключа[4]. **Аутентификация нескольких секторов при помощи одного ключа** После успешной аутентификации состояние регистра не сбрасывается, что делает возможным доступ к данным других секторов[4].

V. РЕАЛИЗАЦИЯ ПРАКТИЧЕСКИХ АТАК

Nested Attack Данная атака использует уязвимости ГПСЧ, утечку битов через код ошибки и parity bits. Достаточно иметь 1 ключ, чтобы восстановить ключи к остальным блокам карты [3]. Пусть известен ключ A(0:0), 0x45a47777d6b3; на рис.5-7 представлены начало, успешное восстановление ключа B(0:0), и последовательное восстановление всех ключей карты соответственно.

```

root@kali: ~ - ~/research/lab/mfoc-mifare-classic-test >> mfoc -k 45a47777d6b3 -o card.mfd
The custom key 0x45a47777d6b3 has been added to the default keys
Found Mifare Classic 1k tag
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 04
+ UID size: single
+ bit frame anticollision supported
  UID (NFCID1): 3a da 67 16
    
```

Рис. 5 – Nested Attack start

```

Sector 00 - Found Key A: 45a47777d6b3 Found Key B: 22609a620491
Sector 01 - Unknown Key A                Unknown Key B
Sector 02 - Unknown Key A                Unknown Key B
    
```

Рис. 6 – Восстановление ключа B блока 0

```

Using sector 00 as an exploit sector
Sector: 1, type A, probe 0, distance 32 .....
Found Key: A [3f9f44338599]
Data read with Key A revealed Key B: [6567787fc991] - checking Auth: OK
Sector: 2, type A, probe 0, distance 32 .....
Found Key: A [4bdf3f676734]
Data read with Key A revealed Key B: [9c5cb713635b] - checking Auth: OK
Sector: 3, type A, probe 0, distance 32 .....
    
```

Рис. 7 – Последовательное восстановление ключей карты

Communication interception attacks

Уязвимость использует 2 ключевых недостатка Crypto1: возможность восстановления состояния LFSR вследствие использования только нечетных бит, что позволяет провести откат состояния регистра до первоначального [3] (рис.8).

```

root@kali: ~ - ~/research/lab/mfoc-mifare-classic-test >> mfoc -k 45a47777d6b3 -o card.mfd
[+] pbs -> NT 144 154
[+] downloading tracelog data from device
[+] Recorded activity (tracelog len = 348 bytes)
[+] starts = start of start frame end = end of frame. src = source of transfer
[+] ISO14443A - all times are in carrier periods (1/13.56MHz)
    
```

Start	End	Src	Data (1 denotes parity error)	CRC	Annotation
992	992	RF	52(7)		MIFA
2244	4612	Tag	04 00		
7040	9504	RF	93 20		ANTICOLL
10020	10036	Tag	3a da 67 16 91		
10072	29600	RF	93 70 3a da 67 16 91 3a a8		ok SELECT_UID
30788	34388	Tag	08 00 00		
35060	40736	RF	09 04 d3 3d		ok AUTH-A(4)
42692	47428	Tag	129 0e 17 94		
50832	69208	RF	02 da 1c 22 061 771 3d b21		ICRC
67360	72664	Tag	071 5a a61 af		
77952	82720	RF	09 02 98 16		ICRC
80036	104808	Tag	03 7a1 b81 af 751 0a1 a9 74 04 23 03 c2 771 5d 51 971 e9 101		ICRC
117688	123656	RF	05 06 461 63		ICRC 7

Рис. 8 – Перехваченные данные

Имеем UID 0x3ada6716, $N_T=0x290e1794$, $\{N_R\}=0x92da1c32$, $\{A_R\}=0x66773db2$, $\{A_T\}=0xb75aa6af$, производим откат состояния регистра и восстанавливаем ключ (рис.9).

```

root@kali: ~ - ~/opt/proxmark3-4rdv4/tools/mfkey >> ./mfkey64 3ada6716 290e1794 92da1c32 66773db2 b75aa6af
MIFARE Classic key recovery - based 64 bits of keystream
Recover key from only one complete authentication!

Recovering key for:
uid: 3ada6716
nt: 290e1794
(nr): 92da1c32
(ar): 66773db2
(at): b75aa6af

LFSR successors of the tag challenge:
nt*: 1651f9e5
nt**: 175fc801

Keystream used to generate (ar) and (at):
ks2: 7026c457
ks3: a095e6ae

Found Key: [3f9f44338599]
    
```

Рис. 9 – Успешное восстановление ключа

VI. ВЫВОДЫ

На основании полученных результатов можно сделать вывод, что использовать теги MIFARE Classic небезопасно. Рассмотренные уязвимости позволяют злоумышленнику восстановить секретный ключ за секунды.

СПИСОК ЛИТЕРАТУРЫ

1. NXP Semiconductors. MIFARE Classic 1K Card IC functional specification (Октябрь 2020)
2. Garcia, Flavio and Gans, Gerhard and Muijers(2008). Dismantling mifare classic. Lect. Note. Comput. Sci.. 5283. 97-114.
3. Garcia, Flavio and van Rossum, Peter and Verdult,(2009). Wirelessly Pickpocketing a Mifare Classic Card. 3-15.
4. Marti Berini Sarrias, Jordi Herrera Joancomarti.Escuela d'Enginyeria. Security in RFID devices, 2013
5. Courtois, Nicolas and Nohl, Karsten and O'Neil, Sean. (2008). Algebraic Attacks on the Crypto-1 Stream Cipher. IACR Cryptology ePrint Archive. 2008. 166.
6. MIFARE Classic EV1 1K- Mainstream Contactless SmartCard IC for fast and easy solution development; Rev. 3.2 - 23.05.2018

Секция «Математическое
моделирование и компьютерная
графика»

Section «Mathematical modeling and
computer graphics»

МОДЕЛИРОВАНИЕ ЭНЕРГЕТИЧЕСКОЙ И ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ТЕПЛОВОЙ МОДЕРНИЗАЦИИ ЖИЛОГО ЗДАНИЯ

Барановский И. В., Тернов Е. В., Башко Ю. А.

Заместитель директора по научной работе, отдел жилищного хозяйства, Институт жилищно-коммунального хозяйства Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: bashko_yury@mail.ru

Тепловая модернизация жилищного фонда Республики Беларусь является актуальным направлением энергосбережения в экономике страны. В рамках перехода отрасли жилищно-коммунального хозяйства (ЖКХ) к ведению хозяйственной деятельности методами цифровой экономики, предполагающего цифровизацию процессов эксплуатации, технического обслуживания и ремонта зданий на принципах BIM-проектирования (Building Information Model), представляется целесообразным проработать методологические вопросы алгоритмизации проектных решений повторного применения, связанных с обеспечением требуемого сопротивления теплопередаче ограждающих конструкций зданий и связанных с ним параметров энергопотребления на цели теплоснабжения и охлаждения.

Введение

Технические нормативные правовые акты Республики Беларусь в области архитектурной, строительной и градостроительной деятельности в значительной степени гармонизированы с европейскими стандартами, а также сводами правил Российской Федерации. Они содержат достаточный методологический аппарат для выполнения теплотехнических и конструктивных расчётов технических решений, направленных на повышение энергоэффективности зданий различного назначения. Вместе с тем остаётся открытым вопрос создания банка цифровых моделей энергопотребления жилых зданий типовых серий постройки 2-й половины XX века, подавляющее большинство которых не соответствует современным требованиям к энергоэффективности. Разработка и унификация указанных моделей позволила бы ускорить и удешевить тепловую модернизацию жилищного фонда страны.

Описание стратегии моделирования энергетической и экономической эффективности здания

Экономико-математическое моделирование здания сводится по существу к разработке математической модели энергетического баланса в системе «здание–окружающая среда», включающей технико-экономическое обоснование технического решения, планируемого к реализации в рамках энергосберегающего мероприятия. При этом принимаются во внимание следующие ключевые факторы [1]:

1. Температура воздуха наиболее холодной пятидневки с обеспеченностью 0,92;
2. Температура воздуха для начала и окончания отопительного сезона и продолжительность отопительного сезона в днях;

3. Месячные суммы прямой и суммарной (прямой, рассеянной и отражённой) солнечной радиации на горизонтальную и вертикальные поверхности различной ориентации по сторонам горизонта при ясном небе и средних условиях облачности;
4. Площадь стен здания за исключением светопроницаемых и наружных дверных проёмов, а также материал и толщина стен как определяющие факторы их термического сопротивления;
5. Конструктивные особенности и термическое сопротивление существующих либо планируемых к установке теплоизолирующих покрытий светонепроницаемых ограждающих конструкций здания;
6. Площади светопроницаемых и наружных дверных проёмов, а также конструктивные особенности и термическое сопротивление окон, витражей, архитектурных фонарей и наружных дверей;
7. Площади покрытий, этажных перекрытий, перекрытий над арками сквозных проходов и техническим подпольем, а также пола в отапливаемом подвале
8. Наличие тамбуров при входных группах, технического подполья либо отапливаемого подвала под перекрытием 1-го этажа, отапливаемого либо не отапливаемого чердака или технического этажа;
9. Вид и технические особенности используемых систем водного (одно- или двухтрубная, централизованная или с индивидуальными регуляторами температуры и др.) или электрического отопления;
10. Вид используемых систем охлаждения и характер инфильтрационных воздушных потоков;

11. Тепловыделение от людей и других источников тепловой энергии в помещениях во временной динамике
12. Требуемая температура воздуха в помещениях здания.

Параметры 1–3 относятся к климатологическим характеристикам географической местности, в которой расположено здания и имеют нормированные значения. Для условий Республики Беларусь значения указанных и иных климатологических характеристик определяются по СНБ 2.04.02-2000 «Строительная климатология». Параметры 4–10 определяются конструктивными особенностями архитектурных элементов ограждающих конструкций, а также инженерных систем здания. Нормативные значения термического сопротивления ограждающих конструкций, методики расчёта термического сопротивления составных теплоизолирующих покрытий стен, окон и наружных дверей представлены в ТКП 45-2.04-196-2010 «Тепловая защита зданий. Теплоэнергетические характеристики. Правила определения». Там же представлены методики расчёта удельного расхода тепловой энергии на отопление здания и тепловыделения в здании на протяжении отопительного периода. Наконец параметры 11–12 отражают характер использования здания в соответствии с его функциональным назначением.

Зарубежным аналогом методического руководства для определения теплоэнергетических характеристик здания является стандарт EN ISO 13790 «Энергетическая эффективность зданий – расчёт потребления энергии для отопления и охлаждения» [2].

С учётом тенденций в архитектурном проектировании к использованию «окон в пол» и светопроницаемых стен при оценке энергоэффективности здания на нужды отопления и охлаждения представляется целесообразным дополнительно оценивать влияние на неё энергозатрат на искусственное освещение и затенение, обеспечивающее нормативные уровни освещённости в различных частях жилого помещения.

Энергетический баланс в системе «здание–окружающая среда» обеспечивается равенством трансмиссионных потерь через ограждающие конструкции здания с учётом инфильтрации окружающего воздуха расходу теплоты на отопление и вентиляцию здания с учётом теплопоступлений от внутренних источников и солнечной радиации.

ЗАКЛЮЧЕНИЕ

Использование электронных архитектурных планов этажей здания упрощает как однозональное, так и многозональное моделирование его теплоэнергетических характеристик с учё-

том процессов теплопередачи через внутренние стены и перегородки в различных частях здания. В перспективе механизм формирования тарифов на отопление может быть пересмотрен по результатам вычислительных экспериментов с многозональными имитационными моделями теплоэнергетических характеристик здания.

Достижение максимальной точности моделирования эксплуатационных показателей здания представляется возможным при экспорте его геометрических характеристик и параметров размещения на местности, теплофизических характеристик материалов и инженерных систем из BIM-модели здания. Использование BIM-модели при моделировании и фактическом осуществлении эксплуатации здания предоставляет следующие преимущества:

- наличие полных и достоверных данных о конструктивных элементах;
- моделирование возможных признаков физического износа конструкций и элементов здания разной степени тяжести;
- детальное планирование технического переоснащения здания;
- мониторинг состояния основных конструктивных элементов здания, возможность отслеживания динамики контролируемых параметров;
- управление энергопотреблением здания;
- оперативная навигация по проекту в случае экстренного устранения последствий внешних ситуаций.

Посредством объединения BIM-модели с расчётами энергопотребления и экономических затрат информационное моделирование позволяет решать вопросы рационального потребления ресурсов отдельным зданием. Ещё на этапе проектирования появляется возможность моделировать ресурсный баланс и эксплуатационный денежный поток, оценивать годовые эксплуатационные доходы и расходы.

СПИСОК ЛИТЕРАТУРЫ

1. Лысёв, В. И. Расчёт энергопотребления для отопления и охлаждения зданий / В. И. Лысёв, Н. Н. Коцюлим, В. А. Кучанский // Научный журнал НИУ ИТМО. Серия «Холодильная техника и кондиционирование» [С.-Пб]. – 2018. – № 1. – С. 3–12.
2. Результаты анализа существующих методологий и практик для осуществления мониторинга интегральных энергетических характеристик зданий в части приложений методов энергетического аудита: отчёт о НИР (ПРООН/ГЭФ) / Проект № 00077154 «Повышение энергетической эффективности жилых зданий в Республике Беларусь»; исп. А. Ф. Молочко. – Минск, 2013. – 15 с. [Электронный ресурс] / Департамент по энергоэффективности Государственного комитета по стандартизации Республики Беларусь. – Режим доступа: <http://energoeffekt.gov.by/effbuild/download/98.pdf>. – Дата доступа: 13.08.2020.

РИСКИ ГЕНЕРАЦИИ ПАРОЛЕЙ

Вельков Д. Е., Фролов Я. И., Гуринович А. Б.

Кафедра вычислительных методов и программирования, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: velkov@list.ru, iaroslav_frolov@mail.ru

Методы безопасного применения генерации случайных чисел в компьютерных технологиях и в банковском деле. Анализ наиболее популярных и безопасных сервисов для генерации паролей.

ВВЕДЕНИЕ

Случайные числа и случайность имеют множество применений в криптографии, науке, играх, искусстве. Для разных задач, требуется генерация разного качества, по этой причине существует потребность в разнообразных методах генерации случайных чисел. Генераторы случайных чисел (ГСЧ) делятся на два основных типа: Генераторы псевдослучайных чисел (ГПСЧ) и Генераторы случайных чисел (ГСЧ). С развитием информационных технологий и интернета, возросла потребность в качественной генерации случайных чисел, не только у специалистов, но и у обычных людей.

I. ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ (ГПСЧ) И ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ (ГСЧ)

Из-за простоты и дешевизны чаще всего используются генераторы, созданные как соответствующие программы на ЭЦВМ. С помощью этих программ по некоторому алгоритму получают последовательности чисел. Алгоритм построен так, что знаки 0 и 1 появляются в среднем одинаковое число раз и отсутствует зависимость между появлениями этих знаков и сформированными из них многозначными числами. Числа получаемые с помощью таких генераторов называются псевдослучайными. Генераторы истинно случайных чисел генерируют последовательности случайных чисел на основе измеряемых, хаотически изменяющихся параметров физического процесса. Работа таких устройств часто основана на использовании надёжных источников энтропии, таких, как тепловой шум, дробовой шум, фотоэлектрический эффект, квантовые явления, погодные явления и другие физические процессы.

II. СИСТЕМА БЕЗОПАСНОСТИ БАНКОВСКИХ КАРТ И АЛГОРИТМЫ ПРОВЕРКИ PIN

PIN-коды для карт могут генерироваться 2 способами. Теперь же перейдём к способу проверки этих самых кодов. На данный момент, в основном, используются следующие 2 алгоритма проверки PIN: Visa PVV и IBM 3624 PIN offset. Visa PVV Данный алгоритм первоначально был

разработан платежной системой Visa, но, в настоящее время является рекомендованным алгоритмом проверки PIN как для карт Visa, так и для MasterCard. В основе данного алгоритма лежит значение PVV (PIN verification value), которое является криптограммой, получаемой на основе следующих величин:

- Номер карты (далее PAN);
- Индекс ключа проверки PIN (PIN verification key index, далее, PVKI);
- Ключ проверки PIN (PIN verification key, далее, PVK)
- Сам PIN код карты.

Для получения PVV формируется блок из PAN (последние 11 цифр, кроме контрольного числа карты), PVKI, PIN (строго, первые 4 цифры), который зашифровывается с помощью PVK, после чего из него, с помощью специальной функции, извлекаются 4-х значное число, которое и является значением PVV [1]. Данное значение PVV является эталонным для проверки PIN кода. Т.е. при получении операции с введенным PIN для его проверки на основании PAN, PVKI, PVK формируется новое значение PVV и сравнивается с эталонным PVV для карты. Если значения совпадают, то PIN считается верным, если не совпадают — неверным. К особенностям данного алгоритма можно отнести следующие «ограничения»:

- Принципиальная невозможность восстановления PIN из значения PVV;
- Использование PIN кода размером строго 4 цифры.

IBM 3624 PIN offset Данный алгоритм первоначально был разработан компанией IBM для использования в банкоматах IBM 3624 В настоящее время данный алгоритм считается устаревшим, но используется по следующим причинам:

- карточные системы «старых» регионов (Западная Европа, Северная Америка) достаточно консервативны и, во многом, работают на «достаточно» старых системах;
- данный алгоритм позволяет восстановить значение PIN кода из проверочного значения.

В основе данного алгоритма лежит значение PIN offset (PIN verification value), которое является

криптограммой, получаемой на основе следующих величин:

- Контрольное значение (Validation data, далее VD) — некоторое значение (обычно — часть номера карты, но это не обязательно);
- Децимализационная таблица (Decimalization table, далее DT);
- Ключ проверки PIN (PIN verification key, далее, PVK);
- Сам PIN код карты.

Для простоты дальнейшего описания под ключом проверки PIN в случае метода IBM 3624 PIN offset будем совокупность ключа PVK и значения таблицы децимализации DT. Для получения PIN offset контрольное значение VD зашифровывается с помощью ключа PVK, после чего из полученного значения с помощью таблицы децимализации DT получается блок из 16 десятичных цифр. Из полученного блока берутся первые N цифр, где N — длина PIN (метод IBM 3624 позволяет проверять PIN с длиной до 16 цифр), далее из каждой цифры PIN по модулю 10 вычитается соответствующая цифра полученного блока. Полученное значение и будет значением PIN offset.

III. ПРОВЕРКА PIN

Терминология

Для упрощения дальнейшего описания введем некоторые термины:

- PIN блок — значение PIN кода карты, упакованной в блок из 8 байт;
- Зашифрованный PIN блок — значение PIN блока, зашифрованное по алгоритму DES/3DES выделенного для целей шифрования PIN блока;
- Проверочное значение PIN — PVV или PIN offset;
- Дополнительные данные проверки PIN — данные, кроме PIN и проверочного значения PIN, необходимые для проверки PIN в соответствии с алгоритмами Visa PVV/IBM 3624 PIN offset.

В части проверки PIN можно указать следующие требования:

- Открытые значения PIN и PIN блока не передаваться, храниться или обрабатываться вне специально отведенных программно аппаратных комплексов;
- Зашифрованный PIN блок — значение PIN блока, зашифрованное по алгоритму DES/3DES выделенного для целей шифрования PIN блока.

Как мы уже определились ранее, для проверки PIN нам необходимы следующие данные:

- Сам PIN, который мы будем проверять;
- Проверочное значение PIN;
- Дополнительные данные проверки PIN.

Открытое значение PIN нельзя получить ни при каких условиях. Имеется доступ только к зашифрованному PIN блоку. В дополнение к нему нужен ключ для его расшифровки. Этот ключ называют РРК (PIN protection key). Имеется два варианта хранения проверочного значения:

Первый вариант — это хранение проверочного значения на магнитной полосе карты после поля Service Code. Второй вариант — это хранение проверочного значения в некотором хранилище, обычно, БД системы, отвечающей за выполнение проверок при авторизации карты.

IV. ЗАКЛЮЧЕНИЕ

Независимо от способа получения информации и получателя, проверка PIN выполняется на HSM, использующий для проверки ключ РРК в защищенном виде, ключ проверки PIN в защищенном виде, зашифрованный PIN блок, проверочное значение PIN и дополнительные данные проверки, в ответ на что возвращается только результат проверки: верный PIN, неверный PIN, прочая ошибка. Т.е. в процессе проверки система, отвечающая за авторизацию, не зависит от открытого значения PIN. Комбинация этих методов обеспечивает минимальные риски при использовании паролей. Для безопасной автоматической генерации паролей целесообразно использовать разнообразные интернет сервисы: Memset, Avast Passwords, ExpressVPN, Norton Password Manager, Random, ClaveSegura, Zoho Vault, Strong Password Generator. Это позволяет точно настроить генерацию, выбрать надёжный пароль и минимизировать риски.

V. СПИСОК ЛИТЕРАТУРЫ

1. Visa Payment Technology Standards Manual, Банковское мошенничество
2. Случайные числа и их применение, Иванова В.М., 1984.
3. <https://www.belinvestbank.by/individual/page/moshennichestvo-v-seti-internet>
4. Feller V. Introduction to the theory of probability and its applications. М.: Мир, 1984.Т.1,2.
5. Gane Samb LO. A Course on Elementary Probability Theory/Statistics and Probability African Society, (SPAS) Books Series.Saint-Louis, Calgary, Alberta. 2016 — ISBN978-2-9559183-3-3, DOI: <http://dx.doi.org/10.16929/sbs/2016.0003-209> p
6. Charles M. Grinstead. Introduction to Probability / Swarthmore College . — 2009. — 520 p.
7. Dimitri P. Bertsekas. Introduction to Probability, lecture notes, Course 6.041-6.431/ Dimitri P. Bertsekas , John N. Tsitsiklis. — M.I.T ,2000— 284 p
8. Kramer G. Mathematical methods of statistics. Moscow: The World, 1976.

МОДЕЛИРОВАНИЕ И РАСЧЕТ ПРОСТРАНСТВЕННОГО РАСПРЕДЕЛЕНИЯ КОНЦЕНТРАЦИИ ИОННО-ИМПЛАНТИРОВАННОЙ ПРИМЕСИ

Гранько С. В., Короткевич А. В.

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: granko@bsuir.by

Предложена модель для расчета объемного распределения концентрации примеси внедренной в твердое тело методом ионной имплантации с использованием маскирующих покрытий. Математической основой модели является использование распределения Пирсона VII, описывающее распределение концентрации при использовании модели точечного пучка ионов. Установлены особенности влияния момента эксцесса на пространственное распределения примеси.

НАЧАЛО

Представляемые в работах [1,2,3] методы построения пространственного распределения концентрации примеси введенной методом ионной имплантации через окно маскирующем покрытии сводятся к использованию описания бокового отклонения, которое претерпевают падающие ионы от направления перпендикулярно поверхности образца с помощью параметра $\Delta R_{p,l}$ – стандартное отклонение боковых пробегов ионов. При этом предполагается, что боковое распределение концентрации имплантированной примеси, или зависимость локальной концентрации от боковой координаты, является гауссовским с боковым проецированным пробегом $R_{p,l}$. Однако при этом пренебрегается моментами более высоких порядков.

Однако при формировании ряда структур методом локального ионного легирования с использованием окон в маскирующем покрытии требуется учет моментов более высокого порядка. При этом пространственное распределение концентрации имплантированной примеси следующее:

$$n(x, y, z) = KDF_1^P(x) \int \int_S F_2^P(Y-y)F_3^P(Z-z)dZdY, \quad (1)$$

где D – доза ионной имплантации;

F_1^P – продольное распределение (по координате x), как правило описывается распределением Пирсона IV;

F_2^P, F_3^P – поперечное распределение по координатам y и z;

S – параметр, характеризующий окно в маскирующем покрытии, имеет размерность площади, определяется координатами Y и Z;

K – нормировочный коэффициент.

Для описания бокового ухода предполагаем, что распределения по продольной координате (функции F_2^P или F_3^P) носят симметричный характер, параметр асимметрии $\gamma=0$ и может быть описано с помощью распределений Пирсона для различных значений эксцесс β . Аналити-

ческие выражения этих функций: определяется следующим образом (на примере координаты y): вводятся следующие переменные зависящие, от моментов распределений:

$$b_0 = \frac{\Delta R_p^2(4\beta - 3\gamma^2)}{10\beta - 12\gamma^2 - 18}, \quad (2)$$

$$b_2 = \frac{2\beta - 3\gamma^2 - 6}{10\beta - 12\gamma^2 - 18}, \quad (3)$$

Распределение Пирсона VII для $\gamma=0$ и $\beta>3$:

$$F^P(y) = \left| y^2 + \frac{b_0}{b_2} \right|^{\frac{1}{2b_2}}, \quad (6)$$

Для $\gamma=0$ и $\beta=3$ имеем распределение Пирсона VIII, которое идентично распределению Гаусса. На рис.1 представлены нормированные относительно максимума распределения Пирсона VII и Пирсона VIII построенные с применением (2-6) для значений асимметрии $\gamma=0$ и эксцесса $\beta_1=3, \beta_2=5, \beta_3=10$. Данные профили распределения относительной концентрации имплантированной примеси могут быть применены в представлении модели точечного пучка ионов попадающего в точку поверхности мишени с координатой $y=0$ мкм (нм, ангстрем).

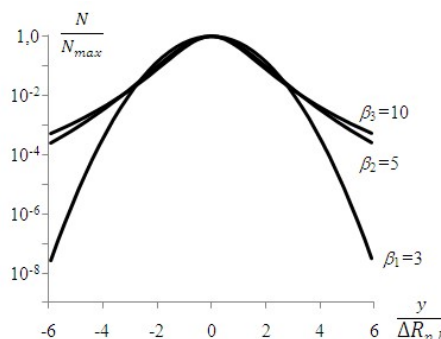


Рис. 1 – Модельные профили поперечного распределения концентрации примеси для точечных пучков

Для расчета пространственного распределения концентрации примеси требуется интегри-

рование данных распределений по площади окна в маскирующем по-крытии. В данной работе проведено моделирование пространственного распре-деления концентрации примеси маскирующей структурой локального ионного легирования является полуплоскость:

$$y \leq 0, \quad (7)$$

На рис. 2. представлены результаты численного интегрирования (1) модельных профилей для полуплоскости. Значения концентрации нормированы относительно максимума, наблюдается уход примеси под маску. На границе раздела «окно-маска» значение концентрации $0,5N_{max}$, что объясняется значением асимметрии $\gamma=0$, боковой уход ионов по обе стороны границы раздела равновероятен. На расстоянии $5\Delta R_{p,l}$ от границы раздела «окно-маска» концентрация примеси сформированной модельным профилем $\beta_2 = 5$ в 10^3 выше сформированной модельным профилем $\beta_1 = 3$. Концентрация примеси сформированной модельным профилем $\beta_2 = 5$ отличается от сформированной модельным профилем $\beta_3 = 10$ незначительно.

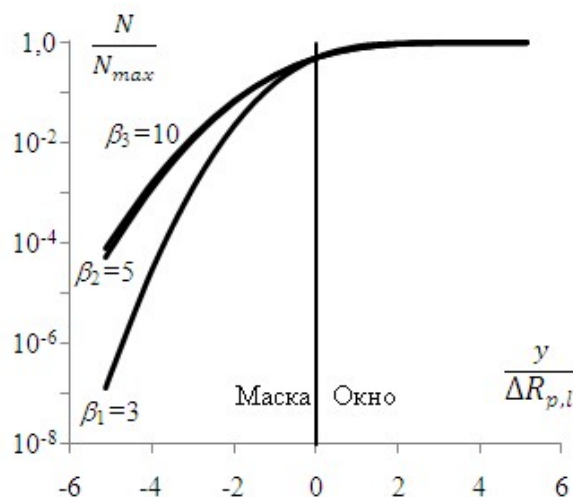


Рис. 2 – Модельные профили поперечного распределения концентрации примеси при использовании маски

Список литературы

1. Комаров Ф.Ф. Ионная и фотонная обработка материалов. - Мн.: Белгосуниверситет, 1998. -209 с.
2. Н. Runge. Distribution of Implanted Ions under Arbitrarily Mask Edges /Phys. Stat. Sol.(a), 1977 .vol.39, pp 595-599,
3. Буренков А.Ф., Комаров Ф.Ф., Темкин М.М. Аналитический расчет ионной имплантации через окно в маске // Микроэлектроника. 1987.- Т.16, №1. с.15-21.

ОЦЕНКА ПАРАМЕТРОВ МОДЕЛИ INAR(1)

Дорошко О. В., Сталевская С. Н.

Кафедра математического моделирования и анализа данных, Белорусский государственный университет
Минск, Республика Беларусь
E-mail: volha.doroshko@gmail.com

В данной работе приведены методы оценки параметров модели INAR(1). Для оценки самих методов введены понятия смещения и вариации оценки. Представленные в данной статье экспериментальные исследования позволяют получить наглядное представление о свойствах полученных оценок.

ВВЕДЕНИЕ

На практике часто встречаются временные ряды, характеристики которых несовместимы с подходом непрерывного моделирования. Целочисленные вариационные временные ряды встречаются во многих контекстах, часто в виде количества событий или отдельных лиц в последовательных интервалах времени. Примерами этого являются число клиентов, ожидающих обслуживания, ежедневное число отсутствующих работников в офисе, число занятых линий телефонной сети, регистрируемое каждый час, число несчастных случаев на производстве каждый месяц, количество сделок с акциями, произведенных за один день и т.д. Для анализа таких рядов в [1] и была введена модель INAR(1).

I. МОДЕЛЬ INAR(1)

Используя, введенный в [2], биномиальный оператор прореживания, вместо обычного оператора умножения, используемого в авторегрессионной AR(1), в [1] и [3] была введена неотрицательная целочисленная авторегрессионная модель INAR(1):

$$X_t = \alpha \circ X_{t-1} + \xi_t \quad (1),$$

где $\alpha \in [0; 1]$ и ξ_t - последовательность независимых и одинаково распределенных неотрицательных целочисленных случайных величин с математическим ожиданием $E(\xi_t) = \mu_\xi$ и дисперсией $V(\xi_t) = \sigma_\xi^2$. Оператор прореживания " \circ " для X - неотрицательной целочисленной случайной величины и любого $\alpha \in [0; 1]$ определяется формулой:

$$\alpha \circ X = \sum_{i=1}^X Y_i,$$

где Y_i - последовательность независимых одинаково распределенных случайных величин Бернулли, не зависящие от X , такие, что

$$Pr(Y_i = 1) = 1 - Pr(Y_i = 0) = \alpha.$$

При этом $\alpha \circ X$ имеет биномиальное распределение с условными параметрами $E(\alpha \circ X | X) = \alpha X$ и $V(\alpha \circ X | X) = \alpha(1 - \alpha)X$ и безусловными параметрами $E(\alpha \circ X) = \alpha\lambda$, где $E(X) = \lambda$, и $V(\alpha \circ X) = \alpha^2\sigma^2 + \alpha(1 - \alpha)\lambda$, где $V(X) = \sigma^2$. Очевидно, что $\alpha \circ X$ принимает только целые значения из $[0; X]$.

II. ОЦЕНКА ПАРАМЕТРОВ МОДЕЛИ

1. Оценка Юле-Уокера.

Для выборки размером n модели $X_t, t \in Z$ автокорреляционная функция равна

$$\hat{p}(k) = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2},$$

где $\bar{X} = \frac{1}{n} \sum_{t=1}^n X_t$ - средняя величина. Оценка Юле-Уокера (Yule-Walker estimator (YW)) для α , основывается на том факте, что $p(k) = \alpha^k$:

$$\hat{\alpha} = \hat{p}(1) = \frac{\sum_{t=1}^{n-1} (X_t - \bar{X})(X_{t+1} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2} \quad (1).$$

Первый момент $X_t, t \in Z$ задается, как $E(X_t) = \frac{\mu_\xi}{1-\alpha}$. Используя это, оценку μ_ξ можно получить следующим образом:

$$\hat{\mu}_\xi = (1 - \hat{\alpha})\bar{X},$$

где $\hat{\alpha}$ берется из (1).

2. Условная оценка методом наименьших квадратов.

В качестве альтернативы метода Юла-Уокера можно рассматривать метод оценки условных наименьших квадратов (Conditional least squares estimation (CLS)). Условная оценка наименьших квадратов $\hat{\eta} = (\hat{\alpha}, \hat{\mu}_\xi)^T$ для $\eta = (\alpha, \mu_\xi)^T$ выглядит следующим образом:

$$\hat{\eta} = \arg \min_{\eta} (S_n(\eta)),$$

где $S_n(\eta) = \sum_{t=2}^n [X_t - g(\eta, X_{t-1})]^2$ и $g(\eta, X_{t-1}) = E(X_t | X_{t-1})$. Таким образом, согласно [4], условная оценка наименьших квадратов для α и μ_ξ может быть представлена, как

$$\hat{\alpha} = \frac{\sum_{t=2}^n X_t X_{t-1} - \frac{\sum_{t=2}^n X_t \sum_{t=2}^n X_{t-1}}{n-1}}{\sum_{t=2}^n X_{t-1}^2 - \frac{(\sum_{t=2}^n X_t X_{t-1})^2}{n-1}} \quad (2)$$

и

$$\hat{\eta}_\xi = \frac{1}{n-1} \left(\sum_{t=2}^n X_t - \hat{\alpha} \sum_{t=1}^n X_{t-1} \right),$$

где $\hat{\alpha}$ из (2).

3. Оценка максимального правдоподобия.

Полагая X_1 фиксированной величиной, условная функция логарифмического правдоподобия выглядит следующим образом:

$$l(\alpha, \mu_\xi) = \log\left(\prod_{t=2}^n Pr(X_t|X_{t-1})\right) = \sum_{t=2}^n \log\left(Pr(X_t|X_{t-1})\right). \quad (1)$$

Оценки методом максимального правдоподобия (conditional maximum likelihood (CML)) $\hat{\alpha}$ и $\hat{\mu}_\xi$ для α и μ_ξ определяются как значения α и μ_ξ , которые максимизируют условную функцию логарифмического правдоподобия в (3).

III. СМЕЩЕНИЕ И ВАРИАЦИЯ ОЦЕНОК

Для изучения свойств построенных оценок будем рассматривать такие характеристики оценки, как смещение и вариация.

Определение

Смещением статистической оценки $b(\hat{\alpha})$ называется уклонение математического ожидания этой оценки от истинного значения параметра α :

$$b(\hat{\alpha}) = E(\hat{\alpha}) - \alpha.$$

Определение

Вариацией (среднеквадратической ошибкой) статистической оценки $v(\hat{\alpha})$ называется следующее математическое ожидание:

$$v(\hat{\alpha}) = E(|\hat{\alpha} - \alpha|^2).$$

Вариация – всегда скалярная величина.

IV. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Чтобы понять относительные достоинства каждого из трех методов оценки, были проведены экспериментальные исследования для разных значений параметров $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$. Значения параметра θ для последовательности μ_ξ приняли равным 1.

CML α была найдена путем максимизации условной функции логарифмического правдоподобия в (3). 200 повторений проводились для каждого из возможных параметров α и были рассчитаны смещение и вариация оценок параметров.

Результаты исследования, представленные в таблицах 1 и 2, показывают соответственно смещение и вариацию при различных значениях α и различных размерах выборки, которые были рассмотрены.

V. СПИСОК ЛИТЕРАТУРЫ

1. First-order integer-valued autoregressive (INAR(1)) process / M. A. Al-Osh, A. A. Alzaid // Journal of Time Series Analysis, 1987. –С. 61–275.
2. Discrete analogues of self-decomposability and stability / F. W. Stuetal, K. Van Harn // The Annals of Probability, 1979. –С. 893-899.
3. Some simple models for discrete variate time series / E. McKenzie // Water Resources Bulletin, 1985. – С. 645–650.
4. On conditional least squares estimation for stochastic processes / L. Klimko, P. Nelson // The Annals of Statistic, 1978. С. 629-642.

Таблица 1 – Смещение оценки параметров

n	α	Смещение(α)			Смещение(θ)		
		Y-W	CLS	CML	Y-W	CLS	CML
100	0.1	-0.0457	-0.0339	0.1127	0.096	0.0708	0.0548
	0.3	-0.0852	-0.07	0.1595	0.1788	0.149	0.0747
	0.5	-0.0889	-0.0705	0.1691	0.2534	0.2142	-0.0572
	0.7	-0.0635	-0.0432	0.001	0.2083	0.1527	0.0213
	0.9	-0.0418	-0.0217	-0.0073	0.2886	0.1828	-0.053
200	0.1	-0.0462	-0.0405	0.0957	0.0774	0.0654	0.0357
	0.3	-0.0727	-0.0651	0.2022	0.1707	0.1572	0.029
	0.5	-0.0808	-0.0721	0.035	0.2196	0.201	0.0453
	0.7	-0.0392	-0.0299	-0.0064	0.1269	0.1022	0.042
	0.9	-0.0209	-0.0106	-0.004	0.1603	0.102	-0.0335

Таблица 2 – Вариация оценки параметров

n	α	Вариация(α)			Вариация(θ)		
		Y-W	CLS	CML	Y-W	CLS	CML
100	0.1	0.0108	0.0104	0.1196	0.0264	0.0236	0.0366
	0.3	0.0153	0.0133	0.1636	0.0537	0.0436	0.0867
	0.5	0.0172	0.014	0.0657	0.0993	0.079	0.0957
	0.7	0.0088	0.0066	0.0091	0.0807	0.0601	0.1364
	0.9	0.0041	0.0022	0.0005	0.287	0.1716	0.0334
200	0.1	0.0066	0.0063	0.1577	0.0151	0.0136	0.0253
	0.3	0.0104	0.0094	0.1308	0.0464	0.0415	0.1014
	0.5	0.0107	0.0093	0.0692	0.0648	0.0561	0.0852
	0.7	0.0051	0.0043	0.004	0.0547	0.0456	0.0303
	0.9	0.0012	0.0008	0.0001	0.0997	0.0723	0.0152

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ УРОВНЕЙ С ИСПОЛЬЗОВАНИЕМ ДВОИЧНОГО РАЗБИЕНИЯ ПРОСТРАНСТВА

Дунин А. Т.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: dunin.vit@gmail.com

В докладе рассматривается возможность применения метода двоичного разбиения пространства для процедурной генерации уровней при разработке roguelike игр.

ВВЕДЕНИЕ

Существует множество способов генерации карт. Естественно, ни один способ не является универсальным решением, поэтому большинство разработчиков подстраивают выбранные ими способы под свои игры.

При заполнении области объектами (например, комнатами в подземелье) в случайном порядке вы рискуете тем, что всё будет слишком случайным. Результат может оказаться абсолютно бесполезным хаосом. Один из способов решения этой проблемы это использование метода двоичного разбиения пространства (Binary Space Partitioning, BSP).

BSP-деревья можно использовать для создания простейших и наиболее характерных для roguelike карт — прямоугольных комнат, соединённых друг с другом коридорами.

I. ОПРЕДЕЛЕНИЕ BSP-ДЕРЕВА

Чтобы составить представление о том, что такое BSP-дерево, взгляните на рис. 1.

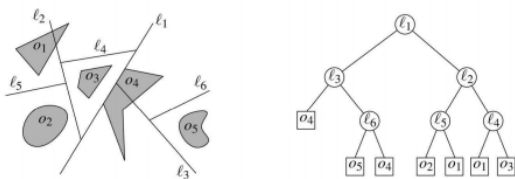


Рис. 1 – Двоичное разбиение пространства и соответствующее дерево

На нем показано двоичное разбиение пространства (BSP) для множества объектов на плоскости, а также соответствующее ему дерево. Как видите, двоичное разбиение пространства получается путем рекурсивного деления плоскости прямой линией: сначала мы делим всю плоскость прямой l_1 , затем делим верхнюю полуплоскость прямой l_2 , а нижнюю - прямой l_3 и так далее. Разделительные прямые не только разбивают плоскость, но могут и разрезать на части объекты. Деление продолжается, пока внутри каждой области не останется только один объект. Этот процесс естественно моделируется двоичным деревом. Каждый лист дерева

соответствует грани окончательного разбиения; часть объекта, попавшая в эту грань, хранится в листе. Внутренние узлы соответствуют разделительным прямым, сама прямая хранится в узле. Если на сцене присутствуют одномерные объекты (отрезки прямых), то может получиться, что объект является частью разделительной прямой, в таком случае эти объекты хранятся во внутреннем узле в виде списка.

Двоичное разбиение пространства, или BSP-дерево для множества S объектов в d -мерном пространстве определяется тогда как двоичное дерево T , обладающее следующими свойствами.

- Если $\text{card}(S) \leq 1$, то T - листовой узел; фрагмент объекта из S (если существует) явно хранится в этом узле. Если обозначить листовой узел v , то множество (возможно, пустое), хранящееся в этом узле, обозначается $S(v)$;
- Если $\text{card}(S) > 1$, то в корне v дерева T хранится гиперплоскость h_v вместе с множеством $S(v)$ объектов, целиком содержащихся в h_v . Левым потомком v является корень BSP дерева T^- для множества $S^- := \{h_v^- \cap s : s \in S\}$, а правым потомком v - корень BSP-дерева T^+ для множества $S^+ := \{h_v^+ \cap s : s \in S\}$.

Размером BSP-дерева называется сумма размеров множеств $S(v)$ по всем узлам v BSP дерева. Иными словами, размер BSP-дерева равен общему числу сгенерированных фрагментов объектов. Если BSP не содержит бесполезных разделительных линии - отсекающих пустое подпространство - то число узлов дерева растет с увеличением размера BSP дерева не быстрее, чем линейно. Строго говоря, размер BSP-дерева ничего не говорит об объеме необходимой для его хранения памяти, поскольку ничего не известно о том, сколько памяти нужно для одного фрагмента. Тем не менее, это хорошая характеристика для сравнения качества различных BSP-деревьев для одного и того же множества объектов.

II. BSP-ДЕРЕВЬЯ И АЛГОРИТМ ХУДОЖНИКА

Допустим, мы построили BSP-дерево T для множества S объектов в трехмерном пространстве. Как с его помощью получить упорядочение по глубине, необходимое для отображения S посредством алгоритма художника? Обозначим P_{view} точку обзора и предположим, что P_{view} расположена выше разделительной плоскости, хранящейся в корне T . Тогда очевидно, что ни один объект, расположенный ниже разделительной плоскости, не может заслонять объекты, расположенные выше нее. Следовательно, мы можем без опаски отображать все объекты (точнее, фрагменты объектов) из поддерева T^- раньше объектов из T^+ . Порядок фрагментов объектов в поддеревьях T^+ и T^- рекурсивно определяется таким же образом. Эффективность любого алгоритма, в котором используются BSP-деревья - существенно зависит от размера BSP-дерева. Поэтому разделительные плоскости нужно выбирать так, чтобы фрагментация объектов была минимальна. Прежде чем переходить к разработке стратегии, порождающей небольшие BSP-деревья, нужно определиться с допустимыми типами объектов. Поскольку скорость стоит на первом месте, типы объектов на сцене должны быть простыми: нужно отказаться от искривленных поверхностей и представлять всё полиэдральной моделью. Таким образом, требуется построить BSP-дерево небольшого размера для заданного множества треугольников в трехмерном пространстве [1].

III. ГЕНЕРАЦИЯ ИЗНАЧАЛЬНОЙ ГЕОМЕТРИИ УРОВНЯ

Алгоритм достаточно прост. Изначально создаём прямоугольник, размером со всё игровое поле (рис 2).

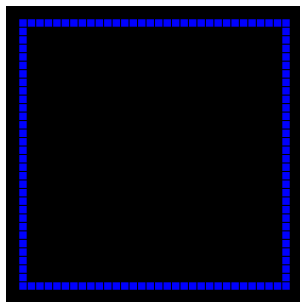


Рис. 2 – Двоичное разбиение пространства и соответствующее дерево

Затем делим его случайным образом на две части — либо горизонтально, либо вертикально.

Где будет проходить линия разделения также выбираем случайным образом (рис. 3).

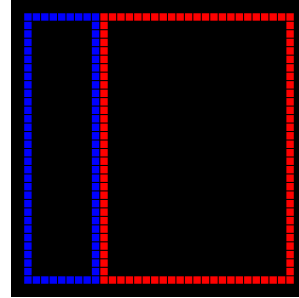


Рис. 3 – Двоичное разбиение пространства и соответствующее дерево

Рекурсивно проделываем тоже самое для новых прямоугольников до некоего предела (рис. 4) [2-3].

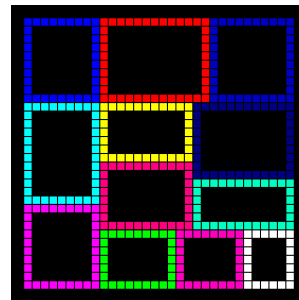


Рис. 4 – Двоичное разбиение пространства и соответствующее дерево

ЗАКЛЮЧЕНИЕ

BSP-деревья находят применение в различных областях, в частности для генерации уровней в играх. Использование процедурных алгоритмов генерации уровней позволяет экономить время на создание сложных закрытых уровней в играх. Также благодаря процедурной генерации можно привнести разнообразие в игры. Современные алгоритмы позволяют создавать уровни неотличимые от уровней созданных человеком.

СПИСОК ЛИТЕРАТУРЫ

1. Mark de Berg. The evaluation and management of bradycardia / Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars//– 2008. – Vol. 386.
2. Информационные системы и сети. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://gamedevelopment.tutsplus.com/tutorials/how-to-use-bsp-trees-to-generate-game-maps-gamedev-12268>. Дата доступа : 18.10.2020.
3. Информационные системы и сети. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.gridsagegames.com/blog/2014/06/procedural-map-generation/>. Дата доступа : 18.10.2020.

ПРИМЕНЕНИЕ АЛГОРИТМА ВЕСОВОГО ОПРЕДЕЛЕНИЯ ГРАНИЧНЫХ ПИКСЕЛЕЙ ДЛЯ ТРЕХКАНАЛЬНОЙ МОДЕЛИ ПЕРЕДАЧИ ЦВЕТА

Заерко Д. В., Липницкий В. А.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: zaerko1991@gmail.com, valipnitski@yandex.by

Предложенный ранее авторами алгоритм весового определения граничных пикселей в алгоритмах масочной фильтрации при обработке растровых изображений в одноканальных системах применяется в случае трех-канальной модели передачи цвета. Рассмотрены особенности применения весового алгоритма для случая большего числа каналов цветопередачи.

ВВЕДЕНИЕ

Наиважнейший этап предварительной обработки изображения – устранение цифрового шума как одного из основных дефектов растрового изображения. Для его успешности широко распространено использование алгоритмов масочной фильтрации (матриц скручивания)[1,2]. Однако сам алгоритм наложения матрицы скручивания имеет свои недостатки. К ним относится работа алгоритма с граничными элементами пиксельной матрицы. Проблема граничных пикселей не нова. Существуют альтернативные методы ее решения, но часто используемые альтернативы не гарантируют соблюдение консенсуса между рациональным использованием вычислительных ресурсов и корректным заполнением граничных пикселей. Одним из методов решения проблемы граничных пикселей является использование весового метода, предложенного в [3,4]. Такой метод может быть также распространен на более сложный случай многоканальности, с сохранением выше упомянутого консенсуса

1. О ВЕСОВОМ АЛГОРИТМЕ ОПРЕДЕЛЕНИЯ ГРАНИЧНЫХ ПИКСЕЛЕЙ.

Базовый алгоритм наложения матрицы скручивания (операция свертки) на пиксельную матрицу предполагает наложение якорного элемента свертки на преобразуемый элемент исходной пиксельной матрицы изображения. Однако при обращении базового алгоритма к граничным пикселям для корректного матричного преобразования, этот алгоритм требует элементы, находящиеся вне границ обрабатываемой пиксельной матрицы. Именно при обращении к таким граничным элементам задействуется весовой метод.

Основная идея весового метода – определение граничных пикселей, исходя из t^2 близлежащих пикселей к проблемному, отсутствующему пикселю. Величина t варьируется в зависимости от размера пиксельной матрицы изображения и от доступных вычислительных ресурсов системы.

В работе [2] приведен пример для матрицы малой размерности и для одноканальной пе-

редачи изображения. В реальных условиях одноканальность (рис 1) уступает место многоканальности, в частности, широко распространенной RGB-модели цветопередачи.

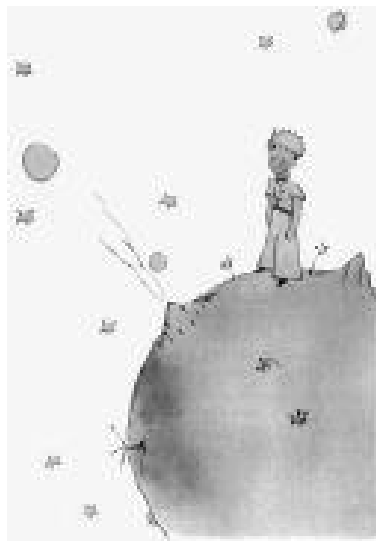


Рис. 1 – Одноканальная передача

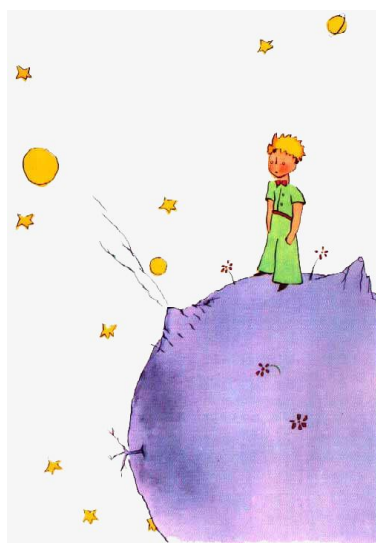


Рис. 2 – Многоканальная передача (RGB)

С точки зрения матричного представления разница заключается в том, что каждый элемент

пиксельной матрицы представлен в виде не одного числового значения в диапазоне $[0, 255]$, а нескольких значений, например, цветовым триплетом для RGB-модели. Он определяет интенсивность цветовых каналов каждого из основных цветов (red, green, blue), которые смешивают для получения составных цветов. Составной цвет задаётся, исходя из упорядоченного набора значений различной интенсивности цветовых каналов. Матрицы для исходного изображения будут иметь следующий вид: M_{mono} - при одноканальной передаче и M_{rgb} - трех-канальной RGB-модели.

$$M_{mono} = \begin{bmatrix} 246 & 246 & \dots & 238 \\ 246 & 246 & \dots & 246 \\ \vdots & \vdots & \ddots & \vdots \\ 246 & 246 & \dots & 245 \end{bmatrix}$$

$$M_{rgb} = \begin{bmatrix} \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 249 \\ 243 \\ 253 \end{bmatrix} \\ \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 247 \\ 244 \\ 255 \end{bmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 245 \\ 245 \\ 243 \end{bmatrix} \end{bmatrix}$$

Матрица описана в сокращенном варианте в связи с ее большой размерностью, даже для изображения с малым разрешением.

II. ПРИМЕНЕНИЕ АЛГОРИТМА ДЛЯ 3-х КАНАЛЬНОЙ МОДЕЛИ ПЕРЕДАЧИ ЦВЕТА. ЧАСТНЫЙ СЛУЧАЙ RGB МОДЕЛИ.

Рассмотрим особенности алгоритма весового определения граничных пикселей для RGB-модели. Легко убедиться в том, что каждый элемент-триплет матрицы для RGB-модели можно разложить на составные части, распределяя полученную часть триплета в отдельную матрицу, соответствующую цветовому каналу из основных цветов RGB-модели: M_{red} , M_{green} , M_{blue} .

$$M_{red} = \begin{bmatrix} 246 & 246 & \dots & 249 \\ 246 & 246 & \dots & 247 \\ \vdots & \vdots & \ddots & \vdots \\ 246 & 246 & \dots & 245 \end{bmatrix}$$

$$M_{green} = \begin{bmatrix} 246 & 246 & \dots & 243 \\ 246 & 246 & \dots & 244 \\ \vdots & \vdots & \ddots & \vdots \\ 246 & 246 & \dots & 245 \end{bmatrix}$$

$$M_{blue} = \begin{bmatrix} 246 & 246 & \dots & 253 \\ 246 & 246 & \dots & 255 \\ \vdots & \vdots & \ddots & \vdots \\ 246 & 246 & \dots & 243 \end{bmatrix}$$

Проведя по-канальное разложение, получаем матрицы определенного вида, для которых возможно применение алгоритма весового определения граничных точек [3,4] одновременно с наложением матрицы скручивания. После проведения масочной фильтрации необходимо покомпонентно, в обратном порядке, составить исходную пиксельную матрицу M_{rgb} , но уже с эффектом наложения матрицы скручивания.

Пусть ядро матрицы скручивания имеет вид K и якорь равен 1, а также оценка граничных пикселей будет определяться, исходя из $t = 3$ близлежащих пикселей. Пиксельная матрица изображения имеет вид M_{rgb} .

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

После выполнения операции свертки можно получить преобразованную матрицу $A_{rgb} = M_{rgb} * K$ следующего вида:

$$A_{rgb} = \begin{bmatrix} \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 249 \\ 243 \\ 253 \end{bmatrix} \\ \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 249 \\ 243 \\ 253 \end{bmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \begin{bmatrix} 246 \\ 246 \\ 246 \end{bmatrix} & \dots & \begin{bmatrix} 243 \\ 245 \\ 240 \end{bmatrix} \end{bmatrix}$$

Последний столбец и строка матрицы указывают на изменения в ходе наложения матрицы скручивания. Алгоритм весового определения граничных пикселей позволил отказаться от фиктивного заполнения элементами и не использовать промежуточных матриц большего размера.

1. Брейсуэлл Р. Н. Преобразование Хартли: Пер с англ. - М.:Мир, 1990. -175 с.
2. Dettmers. T. Understanding Convolutionin Deep Learning / Dettmers. T. // TDBlog, 26.04.2015.
3. Заерко Д. В. Весовой метод решения проблемы граничных пикселей в алгоритме сверточной фильтрации цифрового шума. / Д. В. Заерко, В. А. Липницкий // Кодирование и цифровая обработка сигналов в инфокоммуникациях: материалы международной научно-практической конференции. Минск. - 2020. - С.59 - 63.
4. Заерко Д. В., Липницкий В. А. Алгоритм весового определения граничных пикселей. / Д. В. Заерко, В. А. Липницкий. // Системный анализ и прикладная информатика. - 2020. - № 4. (в печати).

МАТЕМАТИКА ИГРОВОГО БАЛАНСА НА ПРИМЕРЕ ИНКРЕМЕНТАЛЬНЫХ ИГР

Колдаева Е. Ю.

Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: k.koldaeva@gmail.com

В данной работе рассматривается использование методов симуляции игровых систем и поведения игрока для балансировки инкрементальной игры.

ВВЕДЕНИЕ

Игровой баланс является неотъемлемой частью разработки игр, а именно игрового дизайна. То, как настроены параметры каждой из игровых систем и сущностей, определяет опыт, получаемый игроком. Задачей игрового баланса является подбор таких числовых параметров для игровых систем, чтобы итоговый игровой опыт соответствовал ожиданиям гейм-дизайнера.

I. ИНКРЕМЕНТАЛЬНЫЕ ИГРЫ

Инкрементальные игры представляют собой бесконечный экономический цикл, в котором игрок зарабатывает ресурсы, покупает улучшения, начинает зарабатывать еще больше ресурсов. При этом каждое следующее улучшение обходится игроку дороже предыдущего. Центральным элементом инкрементальных игр зачастую является накопление ресурсов, не требующие совершения активных действий со стороны игрока.

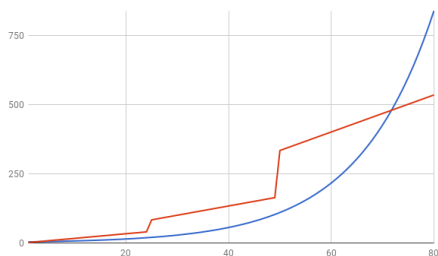


Рис. 1 – Рост цены улучшений и заработка игрока

Рано или поздно игрок достигает точки, в которой прогресс требует непомерных временных затрат. Это происходит потому, что цены улучшений, позволяющих зарабатывать больше ресурсов растут экспоненциально, в то время как заработок игрока является линейной функцией или многочленом.

II. ЭКОНОМИКА ИНКРЕМЕНТАЛЬНОЙ ИГРЫ

С целью декомпозировать экономические системы инкрементальных игр и механики, позволяющие гейм-дизайнерам их контролировать, выделим следующие понятия:

1. Основной ресурс — внутриигровой ресурс, который игрок тратит на покупку улучшений;

2. Дополнительные ресурсы — внутриигровые ресурсы, которые игрок получает в процессе игры и может перевести в основной ресурс;
3. Генератор — внутриигровые сущности, которые прямо или косвенно производят основной ресурс с течением времени и/или при совершении игроком определенных действий;
4. Производные генераторы — внутриигровые сущности, которые вместо основного ресурса производят другие генераторы.



Рис. 2 – Пример цепочки производных генераторов

В инкрементальных играх игрок:

- Накапливает основной ресурс с течением времени;
- Тратит основной ресурс на покупку генераторов;

Стоимость каждого генератора является показательной функцией, которая зависит от количества купленных игроком генераторов. Стоимость каждого следующего генератора рассчитывается по формуле[1]:

$$c_{n+1} = c_0 \times r^n,$$

где c_0 — базовая стоимость генератора, r — основание степени.

Суммарная производительность генераторов каждого типа рассчитывается по формуле[1]:

$$p = p_0 \times n \times,$$

где p_0 — базовая производительность генератора, n — количество генераторов.

Для системы, в которой присутствуют производные генераторы, расчет производительности последнего генератора представляет собой разложение функции e^x в ряд Маклорена:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} > \sum_{n=m}^{\infty} \frac{x^n}{n!},$$

где m — количество «уровней» производных генераторов в игре.

III. БАЛАНСИРОВКА ИНКРЕМЕНТАЛЬНЫХ ИГР

Методом расчета баланса инкрементальной игры было выбрано построение модели игровой экономики, симуляция взаимодействия игрока с этой системой согласно заданному гейм-дизайнером алгоритму поведения, отслеживание численных параметров игровой экономики во времени при такой симуляции и сверка полученных данных с гипотезами и ожиданиями гейм-дизайнера.

Для построения модели абстрактной инкрементальной игры был использован сервис Google Spreadsheets. Для модели поведения игрока была использована стратегия «оптимального выбора», которая подразумевает, что в каждый момент времени игрок стремится потратить ресурсы на генератор с лучшим соотношением стоимости к производительности. В модели были использованы 5 генераторов, производящих основной ресурс. Производительность генераторов задана как единицы основного ресурса в секунду. Параметры генераторов для симуляции взяты из игры «AdVenture Capitalist».

Таблица 1 – Исходные данные симуляции

Генератор	c_0	p_0	r
Генератор 1	4	1.67	1.07
Генератор 2	60	20	1.15
Генератор 3	720	90	1.14
Генератор 4	8640	360	1.13
Генератор 5	103680	2160	1.12

Алгоритм поведения «игрока» работает следующим образом:

1. Алгоритм считает увеличение производительности каждого доступного для покупки генератора (последнего генератора в цепочке для производных генераторов) при условии покупки еще одного генератора.
2. Алгоритм рассчитывает наименьшее соотношение $\frac{c_{n+1}}{p}$, где c_{n+1} — цена следующего генератора, p — производительность генератора после покупки следующего генератора.
3. Алгоритм подсчета времени считает, какое количество времени необходимо игроку для того, чтобы накопить достаточное количество основного ресурса для покупки генератора и добавляет это количество времени к соответствующему счетчику.
4. Алгоритм подтверждает покупку генератора.

Проблема подобной модели в том, что каждый следующий генератор кажется «выгоднее» предыдущего для игрока. Данную проблему

можно решить, добавив коэффициенты к производительности генератора, которые активируются при наличии у игрока достаточного количества генераторов. Суммарная производительность генераторов будет рассчитываться формуле:

$$p = p_0 \times n \times \prod_{i=0}^n k_i,$$

где k_i — дополнительные коэффициенты.

Таблица 2 – Дополнительные коэффициенты

Кол-во	Ген. 1	Ген. 2	Ген.3	Ген. 4	Ген .5
25	3	5	4	2	1.5
50	2	10	8	2	2
100	10	3	4	3	2
200	3	2	2	2	2
...
1000	2	2	2	2	2

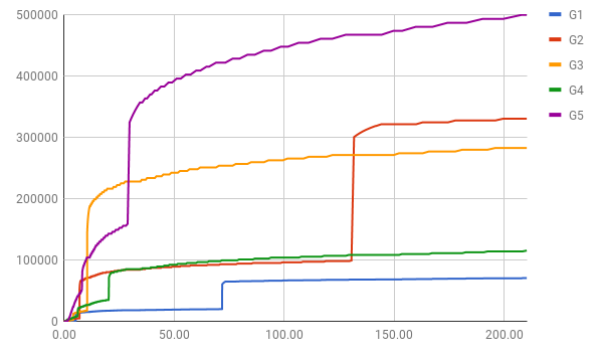


Рис. 3 – Производительность генераторов с учетом дополнительных коэффициентов

ЗАКЛЮЧЕНИЕ

Данная работа иллюстрирует, как методы математического моделирования могут быть использованы в работе гейм-дизайнера, существенно облегчая задачи по расчету игрового баланса.

Созданная в рамках данной работы модель игровой экономики может быть использована для балансировки практически любой инкрементальной игры.

Перспективой для улучшения полученной модели является добавление различных стратегий поведения игрока в симуляцию, добавление производных генераторов, а также дополнительных игровых механик.

1. Kongregate Developers Blog[Electronic resource] /The Math of Idle Games, Part I – Mode of access: <https://blog.kongregate.com/the-math-of-idle-games-part-i/>. – Date of access: 10.11.2020.

МОДЕЛИРОВАНИЕ ГЕНЕРАТОРА ХАОСА НА БАЗЕ КАСКАДНОСВЯЗАННЫХ СИСТЕМ ФАЗОВОЙ СИНХРОНИЗАЦИИ

Кукин Д. П., Гриневич Я. Г., Шатилова О. О.

Кафедра вычислительных методов и программирования, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: kukin@bsuir.by

В докладе рассмотрены вопросы генерирования хаоса устройствами, созданными на основе систем фазовой синхронизации, предложен способ генерации хаотического сигнала, а также устройство генератора, созданного на базе упомянутого выше способа.

ВВЕДЕНИЕ

Принципы фазового управления находят широкое применение в различных областях современной техники. Система фазовой синхронизации (СФС) представляет собой схему, которая эффективно отслеживает разность фаз входного и опорного сигналов [3]. Если разность фаз между входным и подстраиваемым сигналами системы постоянна, то СФС «засинхронизована». Если происходит изменение фазы (частоты) входного или подстраиваемого сигналов, то фазовый детектор в СФС будет вырабатывать сигнал ошибки, пропорциональный величине и полярности изменения фазы. Этот сигнал ошибки вызовет изменение фазы (частоты) опорного сигнала, так что состояние синхронизации вновь восстанавливается. СФС используются для синхронизации генераторов различных частотных диапазонов, включая сверхвысокочастотный диапазон, таким образом, представляется возможным использование хаотических режимов в таких системах для генерации широкополосных сигналов.

В настоящее время в прямохаотических системах связи наибольшее применение получили транзисторные генераторы амплитудно-фазового хаоса. Их особенностью является наличие значительных флуктуации энергии хаотического радиоимпульса, особенно при малых длительностях импульсов. Для фазового хаоса, возникающего в СФС, такие флуктуации выражены значительно меньше, поэтому он является предпочтительным носителем информации для ряда приложений.

1. СПОСОБ ГЕНЕРАЦИИ ХАОТИЧЕСКОГО СИГНАЛА

Основными недостатками широкоизвестных способов генерации широкополосных хаотических сигналов является их техническая сложность [4] и статичность параметров генератора [6]. На рис.1 представлена функциональная схема экспериментальной установки на основе микросхемы 74НС4046А, где I, II, III – фазовые системы, IV – генератор опорных колебаний; k_1, k_2

– величины обратной связи ($0 \leq k_{1,2} \leq 1$), для которой будет проводиться моделирование.

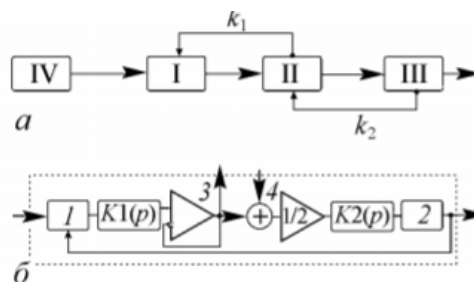


Рис. 1 – Пример ансамбля трех каскадносвязанных фазовых систем (а) и внутренняя структура фазовой системы

Все фазовые системы имеют одинаковую структуру (см. рис 1б): 1 – фазовый дискриминатор; 2 – генератор, управляемый напряжением; $K_1(p), K_2(p)$ – коэффициенты передачи фильтров низких частот первого порядка ($p = d/dt$); 3, 4 – точки выхода/входа сигнала для обеспечения обратной связи между кольцами управления фазовых систем. Коэффициенты передачи фильтров имеют вид $K_{1,2}(p) = 1/(1 + T_{1,2}p)$, где $T_{1,2}$ – постоянные времени фильтров, причем T_1 во всех системах зафиксирован, а состояние T_2 в ходе экспериментов варьировалось, так же как и k_1, k_2, f_1, f_2, f_3 (начальные частоты управляемых напряжением генераторов) и f_0 – частота опорных колебаний. Общий коэффициент передачи двух фильтров в кольце управления $K(p) = 1/(1 + (T_1 + T_2)p + T_1 T_2 p^2)$ соответствует фильтру второго порядка, но при рассматриваемых условиях каждая фазовая система по отдельности обладает исключительно регулярной динамикой. В изолированной фазовой системе возможен либо синхронный, либо квазисинхронный регулярный режим, либо режим биений в зависимости от параметров фильтра и начальной частотной расстройки. Коллективная динамика трех каскадносвязанных фазовых систем качественно отличается от динамики одной системы. В ансамбле возможны различные регулярные и хаотические режимы. Присвоим каждому режиму три индекса $[i_1, i_2, i_3]$. Индекс i_j является 0, если j -я фазо-

вая система находится в синхронном или квазисинхронном режиме (фазовый сдвиг в данной системе фиксирован или изменяется в ограниченных пределах), и равен 1, если j -я фазовая система находится в режиме биений (фазовый сдвиг неограниченно нарастает или убывает). При $k_1 = 0$ и $k_2 = 0$ первая и вторая фазовые системы находятся в синхронном режиме, а третья – в квазисинхронном. При введении обратной связи от третьей системы ко второй появляются колебания в цепи управления второй системы, которые становятся хаотичными при достаточно малом k_3 . При введении обратной связи от второй к первой системе происходит возникновение хаотических модуляций в первой фазовой системе.

Таким образом, управляя величинами связи можно добиться возникновения хаотических колебаний в ансамбле, а также целенаправленно менять режимы хаоса, не меняя параметры элементов. В результате моделирования оказалось, что увеличение коэффициентов обратной связи приводит к увеличению амплитуды и к усложнению хаотических колебаний в кольце управления первой фазовой системы.

На рис. 2-4 представлены спектры хаотических колебаний в первой фазовой системе для различных значений коэффициентов связи k_1 , k_2 .

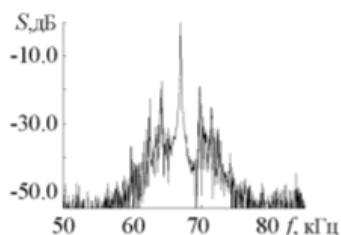


Рис. 2 – Режим $[0,0,1]$, $k_1 = k_2 = 0.2$

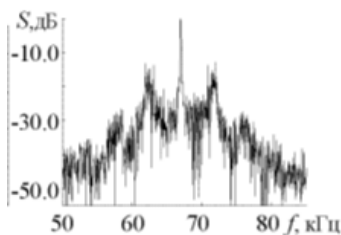


Рис. 3 – Режим $[0,1,1]$, $k_1 = 0.2$, $k_2 = 1.0$

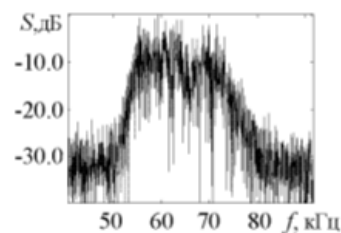


Рис. 4 – Режим $[1,1,1]$, $k_1 = k_2 = 1.0$

II. ЗАКЛЮЧЕНИЕ

Результаты моделирования указывают на то, что ансамбль трех каскадносвязанных фазовых систем позволяет генерировать хаотические колебания в широких и однородных областях параметров. Управление хаотическими режимами с различными спектральными и корреляционными свойствами может осуществляться с помощью изменения величин связей без изменения внутренних параметров элементов ансамбля. Зафиксировано, что сигнал, соответствующий режиму хаотических биений, может обладать более равномерным и широким спектром, более резко спадающей автокорреляционной функцией в сравнении с сигналом, который соответствует квазисинхронному хаотическому режиму. Следовательно, режим хаотических биений также может быть использован в качестве несущих в широкополосных системах связи с некогерентным приемом.

III. СПИСОК ЛИТЕРАТУРЫ

1. Дмитриев А. С., Панас А. И. Динамический хаос: новые носители информации для систем связи. М.: ФМЛ, 2002. 252 с.
2. Залогин Н. Н., Кислов В. В. Широкополосные хаотические сигналы в радиотехнических и информационных системах. М.: Радиотехника. 2006. 208 с.
3. Шахильдян В. В., Ляховкин А. А. Системы фазовой автоподстройки частоты, М.: Связь, 1972.
4. Korzinova M. V., Matrosov V. V. and Shalfeev V. D. Communications using cascade coupled phase-locked loop chaos // Int. J. Bifurcation and Chaos. 1998. Vol. 9, № 5. P. 963.
5. Мишагин К. Г., Матросов В. В., Шалфеев В. Д., Шохнин В. В. Экспериментальное исследование генерации хаотических колебаний в ансамбле двух каскадно-связанных фазовых систем // Письма в ЖТФ. 2005. Т. 31, вып. 24. С. 31.4
6. «Способ генерирования широкополосных свч хаотических сигналов и генератор широкополосных свч хаотических сигналов», Н03В29/00, 12.04.2005, пат. RU 2327278 С2.
7. Дмитриев А. С., Широков М. Е. "Выбор генератора для прямохаотической системы связи Радиотехника и электроника, 2004, т. 49, №7, с. 840-849.
8. Gardner F. M. Charge-Pump Phase-Lock Loops. // IEEE Transactions on Communications. Vol. com-28, №11 November, 1980, p. 1849-1858, пат. US 5055803.
9. Кукин, Д. П. Математическое описание многокольцевых систем фазовой синхронизации / Д. П. Кукин, И. Л. Свито // Информационные технологии и системы 2014 (ИТС 2014) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 29 октября 2014 г. – Information Technologies and Systems 2014 (ITS 2014) : Proceeding of The International Conference, BSUIR, Minsk, 29th October 2014 / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2014. – С. 64–65.
10. Шилин, Л. Ю. Детерминированный хаос в передаче информации / Л. Ю. Шилин, Д. П. Кукин, Н. С. Жилач // Информационные технологии и системы 2017 (ИТС 2017) = Information Technologies and Systems 2017 (ITS 2017) : материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2017 года) / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2017. – С. 64 - 65.

К МЕТОДАМ ПОСТРОЕНИЯ НАЧАЛЬНЫХ ДОПУСТИМЫХ РЕШЕНИЙ ЭКСТРЕМАЛЬНЫХ ЗАДАЧ СЕТЕВОЙ ОПТИМИЗАЦИИ

Пилипчук Л. А., Полячок Е. Н.

Кафедра компьютерных технологий и систем, УО «Белорусский государственный университет»

Минск, Республика Беларусь

E-mail: pilipchuk@bsu.by

Предложены новые подходы и стратегии построения начальных допустимых решений для сетевых моделей математического программирования. На основе исследования теоретико-графовых свойств базисов экстремальных задач об оптимальных путях и максимальном потоке предложены эффективные вычислительные алгоритмы и технологии построения начальных допустимых решений с применением аппарата теории графов, разреженного матричного анализа и современных достижений теоретической информатики.

ВВЕДЕНИЕ

В конструктивной теории математического программирования актуальной задачей является обоснование алгоритмических подходов при создании методов построения начальных допустимых решений задач сетевой оптимизации. Для экстремальных задач об оптимальных путях и максимальном потоке предложены новые подходы и стратегии построения начальных допустимых решений. Разработка эффективных алгоритмов основана на исследовании теоретико-графовых свойств сетевых моделей, анализе их вычислительной сложности, применении результатов теоретической информатики и на новых возможностях организации вычислительных процессов, крупных изменениях используемых вычислительных средств, появлении более эффективных языков программирования и многопроцессорных систем.

I. НАЧАЛЬНОЕ ДОПУСТИМОЕ РЕШЕНИЕ

Математическая модель задачи о кратчайших путях из узла $s \in I$ в узлы $i \in I \setminus \{s\}$ конечного ориентированного связного графа $G = (I, U)$ с множеством узлов I и множеством дуг U , определенных на $I \times I$, $|I| < \infty$, $|U| < \infty$ имеет следующий вид:

$$\sum_{(i,j) \in U} c_{i,j} x_{i,j} \rightarrow \min \quad (1)$$

$$\sum_{j \in I_i^+(U)} x_{i,j} - \sum_{j \in I_i^-(U)} x_{j,i} = \begin{cases} n-1, & i = s, \\ -1, & i \in I \setminus \{s\}, \end{cases} \quad (2)$$

$$0 \leq x_{i,j} \leq n, (i,j) \in U, n = |I|, x_{i,j} \in \mathbb{N}, \quad (3)$$

где $I_i^+(U) = \{j \in I : (i,j) \in U\}$, $I_i^-(U) = \{j \in I : (j,i) \in U\}$, $x_{i,j}$ – величина дугового потока дуги (i,j) , $c_{i,j}$ – стоимость перемещения по дуге (i,j) единицы потока $x_{i,j}$ из узла i в узел

j , $x = (x_{i,j}, (i,j) \in U)$ – вектор дуговых потоков. Дуговой поток $x_{i,j}$ любой дуги графа G не превышает $n-1$ единиц, т.е. ограничения $x_{i,j} \leq n$, $(i,j) \in U$ из (8) являются излишними. Ограничения (8) примут вид:

$$x_{i,j} \geq 0, (i,j) \in U, x_{i,j} \in \mathbb{N}. \quad (4)$$

Матрица условий сохранения потока (2) является матрицей инцидентности графа G . Матрица инцидентности графа абсолютно унимодулярная, что гарантирует целочисленность оптимального потока x экстремальной задачи (1), (2), (4). Итак, экстремальная задача (1), (2), (4) состоит в том, чтобы минимизировать общее расстояние, пройденное потоком величины $n-1$ единиц из узла s до всех узлов $i \in I \setminus \{s\}$, при этом в каждом узле $i \in I \setminus \{s\}$ требуется одна единица потока.

Пара $\{x, G_0\}$ – опорный поток, где $x = (x_{i,j}, (i,j) \in U)$ – допустимый поток (выполняются ограничения (2), (4)) и $G_0 = (I, U_0)$ – произвольное покрывающее дерево, $U_N = U \setminus U_0$. Базис пространства решений однородной системы, порожденной из системы (2), составляют характеристические векторы $\delta(\tau, \rho) = (\delta_{ij}^{\tau\rho}, (i,j) \in U)$, $(\tau, \rho) \in U \setminus U_0$. Эффективные алгоритмы построения базиса пространства решений $\{\delta(\tau, \rho), (\tau, \rho) \in U \setminus U_0\}$ приведены в [1, 2]. Покрывающее дерево $G_0 = (I, U_0)$ совпадает с деревом достижимости графа G (существует единственный путь из s в каждый узел $i \in I \setminus \{s\}$). Для узлов $I \setminus \{s\}$ и дуг U_0 дерева достижимости графа G построим биективное отображение узлов $i \in I \setminus \{s\}$ на соответствующие элементы $\{\text{pred}(i), \forall i \in I \setminus \{s\}\}$, где $\text{pred}(i)$ есть отец узла i в корневом дереве, $\text{pred}(s) = 0$. Полученное в результате биекции ориентированное дерево $G_0 = (I, U_0)$ называется корневым деревом графа G с корнем в узле s . Дерево достижимости графа G совпадает с корневым деревом.

Приведем списковые структуры [2 – 4], необходимые для реализации базовых опера-

ций с элементами корневых деревьев. Список $\{\text{pred}[i], i \in I\}$ определяет для каждого узла $i \in I$ значение $|\text{pred}[i]|$, которое содержит элемент биективного отображения узла i корневого дерева; знак минус определяет корень узла корневого дерева и может содержать число узлов корневого дерева или другую информацию. Элемент списка $\{d[i], \forall i \in I\}$ определяет направление дуги в корневом дереве $G_0 = (I, U_0)$: если узел i является корнем, то $d[i] = 0$; если дуга $(|\text{pred}[i]|, i) \in U$, то $d[i] = 1$; если дуга $(i, |\text{pred}[i]|) \in U$, то $d[i] = -1$. Список $\{\text{depth}[i], \forall i \in I\}$ определяет для каждого узла $i \in I$ уровень узла i : длину (в дугах) до корня в корневом дереве. Элементы династического обхода [2, 3, 4] корневого дерева хранятся в списке $\{\text{thread}[i], \forall i \in I\}$.

Пусть задано некоторое корневое дерево $G_0 = (I, U_0)$ с корнем в узле s и список $\{\text{pred}[i], i \in I\}$, который определяет для каждого узла $i \in I \setminus \{s\}$ значение предка $|\text{pred}[i]|$ (элемент биективного отображения узла i корневого дерева), $\text{pred}[s] = 0$. Построим начальное допустимое решение $x = (x_{i,j}, (i,j) \in U)$ экстремальной задачи (1), (2), (4). Для дуг опорного потока $\{x, G_0\}$, не входящих в состав дерева $G_0 = (I, U_0)$, величины дуговых потоков положим равными нулю: $x_{i,j} = 0, (i,j) \in U_N = U \setminus U_0$.

Теорема 1. Для узла $i \in I \setminus \{s\}$ величина дугового потока $x_{\text{pred}[i],i}$ каждой дуги $(\text{pred}[i], i) \in U_0$ равна количеству узлов поддерева с корнем в узле i .

Обозначим m_i – число узлов поддерева с корнем в узле i .

Теорема 2. Теоретическая оценка сложности алгоритма нахождения последовательности узлов поддерева с корнем в узле i , основанная на применении структур данных: $\{\text{thread}[i], \forall i \in I\}$ и $\{\text{depth}[i], \forall i \in I\}$ равна $O(m_i), \forall i \in I$.

Математическая модель поиска кратчайшего пути между двумя узлами s и t графа $G = \{I, U\}$ имеет следующий вид:

$$\sum_{(i,j) \in U} c_{i,j} x_{i,j} \rightarrow \min \quad (5)$$

$$\sum_{j \in I_i^+(U)} x_{i,j} - \sum_{j \in I_i^-(U)} x_{j,i} = \begin{cases} 1, & i = s, \\ -1, & i = t, \\ 0, & i \in I \setminus \{s, t\} \end{cases} \quad (6)$$

$$x_{i,j} \in \{0, 1\}, (i,j) \in U \quad (7)$$

Построим начальное допустимое решение задачи (5)–(7). Величины дуговых потоков для дуг, кроме дуг единственного пути из s в t положим равными нулю. Дуговые потоки дуг единственного пути из узла s в узел t в корневом дереве, которое соответствует покрывающему дереву $G_0 = (I, U_0)$, положим равными 1.

Математическая модель задачи определения *максимального потока* v в сети $G = (I, U)$ имеет следующий вид:

$$v \rightarrow \max, \quad (8)$$

$$\sum_{j \in I_i^+(U)} x_{i,j} - \sum_{j \in I_i^-(U)} x_{j,i} = \begin{cases} v, & i = s, \\ -v, & i = t, \\ 0, & i \in I \setminus \{s, t\} \end{cases} \quad (9)$$

$$0 \leq x_{i,j} \leq d_{i,j}, (i,j) \in U, \quad (10)$$

где $I_i^+(U) = \{j \in I : (i,j) \in U\}$,

$I_i^-(U) = \{j \in I : (j,i) \in U\}$.

Матрица системы (9) – абсолютно унимодулярная, что гарантирует целочисленность оптимального решения $\{x, v\}[2]$.

Начальное допустимое решение (x, v) экстремальной задачи (8)–(10) сформируем следующим образом. В сети $G = (I, U)$ построим произвольный путь $L_{s,t}$ из узла s в достижимый узел t . Положим $x_{i,j} = 0, (i,j) \in U \setminus L_{s,t}$. Дуговые потоки дуг пути $L_{s,t}$ положим равными величине пропускной способности $d(L_{s,t})$ пути $L_{s,t}$:

$$d(L_{s,t}) = \min \{d_{i,j}, (i,j) \in L_{s,t}\}.$$

II. ЗАКЛЮЧЕНИЕ

Предложены новые подходы и стратегии построения начальных допустимых решений для экстремальных задач поиска оптимальных путей и задачи о максимальном потоке. На основе исследования комбинаторных свойств сетевых моделей, анализе их вычислительной сложности, применении результатов теоретической информатики, аппарата теории графов, разреженного матричного и сетевого анализа и новых возможностей организации вычислительных процессов предложены эффективные методы построения начальных допустимых решений. Разработанные структурные, алгоритмические и технологические решения могут быть применены в теории математического программирования при создании базисных методов построения оптимальных решений задач потокового программирования.

СПИСОК ЛИТЕРАТУРЫ

1. Pilipchuk, L.A. The general solutions of sparse systems with rectangular matrices in the problem of sensors optimal location in the nodes of a generalized graph / L.A. Pilipchuk, O.V. German, A.S. Pilipchuk // Вестник БГУ. Серия 1. Физика. Математика. Информатика. – 2015. – №2. С. 91 – 96.
2. Pilipchuk, L.A. Sparse Linear Systems and Their Applications / L.A. Pilipchuk. – Minsk: BSU, 2013. – 235 p.
3. Ahuja, R.K. Network Flows: Theory, Algorithms, and Applications / R.K. Ahuja, T.L. Magnanti, J.B. Orlin. – New Jersey: Prentice Hall, 1993. – 864 p.
4. Йенсен, П. Потокоевое программирование / П. Йенсен, Д. Барнес. – М.: Радио и связь, 1984. – 392 с.

О ЧИСЛЕННЫХ МЕТОДАХ И ТЕХНОЛОГИЯХ ПОСТРОЕНИЯ КРАТЧАЙШИХ ПУТЕЙ

Пилипчук Л. А., Полячок Е. Н., Лобко А. Э., Шкурский Д. А.

Кафедра компьютерных технологий и систем, УО «Белорусский государственный университет»

Минск, Республика Беларусь

E-mail: pilipchuk@bsu.by, artem1102zx@gmail.com, daniilshkurski@gmail.com

Работа посвящена исследованию эффективности алгоритмов поиска кратчайших путей в графе с использованием новых возможностей организации вычислительных процессов, заложенных в высокоуровневых языках программирования, которые имеют в наличии дополнительные структуры и библиотеки для прикладного использования. Применяются новые технологии разреженного матричного и сетевого анализа в синтезе с современными достижениями в области теоретической информатики.

ВВЕДЕНИЕ

В различных технических, экономических, информационных, экологических, транспортных и других системах актуальной проблемой является создание эффективных вычислительных методов, алгоритмов и технологий для решения задачи поиска оптимальных путей в графе. Крупные изменения используемых вычислительных средств, улучшение характеристик вычислительной техники, ее быстродействия и оперативной памяти, появление многопроцессорных систем и новых возможностей организации вычислительных процессов, заложенных в высокоуровневых языках программирования, которые имеют в наличии дополнительные структуры и библиотеки для прикладного использования, позволяют создать эффективные вычислительные алгоритмы построения оптимальных (субоптимальных) решений задачи об оптимальных путях с использованием методов динамического программирования [1] и базисных методов [2], основанных на концепциях симплекс-метода, с использованием результатов теории графов, разреженного матричного анализа в синтезе с современными достижениями в области теоретической информатики.

1. ПРИНЦИП ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ЗАДАЧИ О КРАТЧАЙШЕМ ПУТИ. УРАВНЕНИЕ БЕЛЛМАНА

Задан конечный ориентированный связный граф $G = (I, U)$ с множеством узлов I и множеством дуг U . Обозначим $I_i^+(U) = \{j \in I : (i, j) \in U\}$, $I_i^-(U) = \{j \in I : (j, i) \in U\}$, $x_{i,j}$ – величина дугового потока дуги (i, j) , $c_{i,j}$ – стоимость перемещения единицы дугового потока $x_{i,j}$ по дуге (i, j) , $c(L_{s,t})$ – стоимость пути $L_{s,t}$ из узла s в достижимый узел t ,

$$c(L_{s,t}) = \sum_{(i,j) \in L_{s,t}} c_{i,j} x_{i,j}.$$

Математическая модель задачи поиска кратчайшего пути из узла s в достижимый узел

t графа $G = (I, U)$ имеет следующий вид:

$$\sum_{(i,j) \in U} c_{i,j} x_{i,j} \rightarrow \min \quad (1)$$

$$\sum_{j \in I_i^+(U)} x_{i,j} - \sum_{j \in I_i^-(U)} x_{j,i} = \begin{cases} 1, & i = s, \\ -1, & i = t, \\ 0, & i \in I \setminus \{s, t\}, \end{cases} \quad (2)$$

$$x_{i,j} \geq 0, (i, j) \in U, \quad x_{i,j} \in \mathbb{N}, \quad (3)$$

Построим начальное допустимое решение задачи (1) – (3). Величины дуговых потоков для дуг произвольного пути $L_{s,t}$ из узла s в узел t положим равными единице. Дуговые потоки остальных дуг графа $G = (I, U)$ положим равными 0. Стоимость $c(L_{s,t})$ пути $L_{s,t}$ из узла s в достижимый узел t равна

$$c(L_{s,t}) = \sum_{(i,j) \in L_{s,t}} c_{i,j} x_{i,j} = \sum_{(i,j) \in L_{s,t}} c_{i,j}.$$

Длину кратчайшего пути из узла s в произвольный достижимый узел i обозначим B_i (функция Беллмана). Согласно принципу динамического программирования [1] функция B_i удовлетворяет уравнению Беллмана

$$B_j = \min_{i \in I_j^-(U)} \{B_i + c_{i,j}\}, \quad j \neq s, \quad j \in I \quad (4)$$

с краевым условием

$$B_s = 0 \quad (5)$$

при условии, что в графе $G = (I, U)$ нет отрицательных контуров, $I_j^-(U) = \{i : (i, j) \in U\}$.

Если $B_i, i \in I$ является решением уравнения Беллмана (4) с краевым условием (5) и $\{i, i_k, i_{k-1}, \dots, i_2, s\}$ – последовательность узлов, найденная из соотношений

$$B_i = B_{i_k} + c_{i_k, i} = \min_{j \in I_i^-(U)} (B_j + c_{j,i}),$$

$$B_{i_k} = B_{i_{k-1}} + c_{i_{k-1}, i_k} = \min_{j \in I_{i_k}^-(U)} (B_j + c_{j, i_k}),$$

...

$$B_{i_2} = B_s + c_{s,i_2} = \min_{j \in I_{i_2}^-(U)} (B_j + c_{j,i_2}),$$

то последовательность дуг

$$(s, i_2), (i_2, i_3), \dots, (i_k, i)$$

составляет кратчайший путь из узла s в узел i .

II. БАЗИСНЫЕ РЕШЕНИЯ ЗАДАЧИ О КРАТЧАЙШИХ ПУТЯХ

Математическая модель задачи поиска кратчайших путей из узла s в достижимые узлы $I \setminus \{s\}$ графа $G = (I, U)$ имеет следующий вид:

$$\sum_{(i,j) \in U} c_{i,j} x_{i,j} \rightarrow \min \quad (6)$$

$$\sum_{j \in I_i^+(U)} x_{i,j} - \sum_{j \in I_i^-(U)} x_{j,i} = \begin{cases} n-1, & i = s, \\ -1, & i \in I \setminus \{s\}, \end{cases} \quad (7)$$

$$x_{i,j} \geq 0, (i,j) \in U, n = |I|, x_{i,j} \in \mathbb{N}, \quad (8)$$

$$I_i^+(U) = \{j \in I : (i,j) \in U\}, \quad I_i^-(U) = \{j \in I : (j,i) \in U\}.$$

Экстремальная задача (6) – (8) состоит в том, чтобы минимизировать общее расстояние, пройденное потоком величины $n-1$ единиц из узла s до всех узлов $I \setminus \{s\}$, при этом в каждом узле $I \setminus \{s\}$ требуется одна единица потока, $n = |I|$.

Пусть задано начальное базисное множество дуг U_0 , представляющее собой покрывающее дерево $G_0 = (I, U_0)$, при этом, G_0 – *дерево достижимости* из узла s (для $\forall i \in I \setminus \{s\}$ существует единственный путь из узла s). Следуя [2] построим начальное допустимое решение задачи (1) – (3) [2]. Опорный поток $\{x, U_0\}$ является невырожденным, так как для дуговых потоков дуг корневого дерева выполняются условия $x_{i,j} > 0, (i,j) \in U_0$, при этом, допустимые решения $x = (x_{i,j}, (i,j) \in U)$ экстремальной задачи (6) – (8) являются целочисленными.

Подсчитаем оценки $\Delta_{\tau\rho}, (i,j) \in U_N = U \setminus U_0$

$$\begin{aligned} \Delta_{\tau\rho} &= c_{\tau\rho} + \sum_{(i,j) \in U_0} c_{ij} \delta_{ij}^{\tau\rho} = \sum_{(i,j) \in L(\tau,\rho)} c_{ij} \delta_{ij}^{\tau\rho} = \\ &= c_{\tau\rho} - (u_\tau - u_\rho), (\tau, \rho) \in U_N, \end{aligned}$$

где $u_i, i \in I$ – потенциалы, которые вычисляются из системы:

$$u_i - u_j = c_{i,j}, (i,j) \in U_0, u_s = 0.$$

В [2] представлены структурные и алгоритмические решения задачи (6) – (8) построения

оптимальных путей в графе с использованием технологий разреженного матричного и сетевого анализа в синтезе с современными достижениями в области теоретической информатики.

При разработке конструктивной теории решения экстремальной задачи поиска кратчайших путей из заданного узла во все достижимые узлы особое внимание уделено специфике корневых деревьев и эффективной реализации поиска наименьшего общего предка при построении фундаментального цикла, удалению дуги из дерева и слиянию корневых деревьев. С применением результатов теоретической информатики разработаны алгоритмы и структуры данных для хранения и преобразования корневых деревьев. Разработан эффективный алгоритм построения начального допустимого решения с использованием алгоритма поиска поддерева с корнем в заданном узле. Для повышения эффективности базисного метода в силу специфики условий оптимальности для допустимого опорного потока $\{x, U_0\}$ и свойств корневого дерева U_0 совокупность операций $\{x, U_0\} \rightarrow \{\tilde{x}, \tilde{U}_0\}$ итерационного процесса сводится преобразования корневых деревьев $U_0 \rightarrow \tilde{U}_0$. В результате структурных преобразований корневых деревьев получено дерево кратчайших путей \tilde{U}_0 (выполняются условия оптимальности). Дуговой поток $\tilde{x}_{pred[i],i}$ дуги $(pred[i], i)$ равен числу узлов поддерева с корнем в узле $i \in I$ для оптимального решения \tilde{x} [2].

III. ЗАКЛЮЧЕНИЕ

Рассматриваются методы построения оптимальных решений задачи об оптимальных путях с использованием методов динамического программирования [1] и базисных методов [2]. При создании вычислительных методов построения оптимальных решений задач о кратчайших путях, особое внимание уделяется разработке алгоритмических, структурных и технологических решений, основанных на современных достижениях теоретической информатики и результатах разреженного матричного и сетевого анализа

СПИСОК ЛИТЕРАТУРЫ

1. Романовский И. В. Перебор субоптимальных решений в дискретных задачах оптимизации / И. В. Романовский // Компьютерные инструменты в образовании. – 6, 2012. – С. 25 – 34.
2. Пилипчук, Л.А. Оптимальные пути: алгоритмические, структурные и технологические решения / Л.А. Пилипчук [и др.] // Веснік Гродзенскага дзяржаўнага ўніверсітэта імя Янкі Купалы. Серыя 2. Матэматыка. Фізіка. Інфарматыка, вылічальная тэхніка і кіраванне. – Т 10 (3), 2020. – С. 143 – 151.

ТЕСТИРОВАНИЕ ИНТЕРФЕЙСА МОБИЛЬНЫХ ИГР

Потапчик В. С.

Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: potapchik97@gmail.com

Рассматривается процесс тестирования интерфейса мобильных игр и основные технологии тестирования интерфейса.

ВВЕДЕНИЕ

Тестирование мобильных игр и приложений – это отдельное направление в обеспечении качества программного обеспечения. Тестирование интерфейса игр сводится к имитации действий пользователя, задача тестирования – убедиться, что все компоненты системы правильно взаимодействуют друг с другом. Сейчас существуют несколько различных технологий для тестирования интерфейса игр и приложений, поэтому возникает необходимость в подборе корректного инструмента для решения задач тестирования.

I. ПРОЦЕСС ТЕСТИРОВАНИЯ ИНТЕРФЕЙСА МОБИЛЬНЫХ ИГР

В момент тестирования пользовательского интерфейса осуществляется проверка, насколько он (интерфейс) удобен в использовании, соответствует ли требованиям, как ведет себя игра и отображаются ли элементы пользовательского интерфейса на различных устройствах при выполнении пользователем определенных действий. Это позволяет оценить насколько пользователю удобно работать с приложением

Целесообразно проводить тестирование интерфейса на начальном этапе разработки мобильного приложения, на этапе прототипа. Одновременно с тестированием интерфейса проводится и тестирование впечатления пользователя, то есть определяется, как человек себя чувствует при взаимодействии с системой. Тестирование интерфейса помогает проверить большую часть действий пользователя, взаимодействие сервисов и компонентов.

На данном этапе выявляются структурные и визуальные недостатки интерфейса игр, проверяется возможность полного функционирования приложения. В ходе проверки акцентируется внимание на то, каким образом элементы интерфейса реагируют на действия пользователя, как приложение обрабатывает эти действия, зачастую это нажатия на экран мобильного устройства. Одной из ключевых проверок является соответствие пользовательского интерфейса стандартным разрешениям смартфонов, таких разрешения как: 1366x768, 1400x900, 1680x1050, 1280x720, 2560x1440, 2960x1440, 2880x1440.

Не всегда устройства с необходимым разрешением доступны специалисту по тестированию. Такие ситуации решает использование эмуляторов. Эмулятор – это программа, которая позволяет имитировать функции другого мобильного телефона, который вы хотите имитировать, устанавливая их на свой компьютер или мобильный телефон. Современные эмуляторы стремятся имитировать не только программное обеспечение мобильных устройств, но и их физические характеристики. Таким образом в ходе тестирования интерфейса мобильных игр используется сочетание реальных устройств и эмуляторов, что позволяет оптимально и максимально подробно покрыть тест-кейсы.

После окончания тестирования предоставляется отчет о тестировании – документ, подводящий итог задачам и результатам тестирования, также содержащий описание проведенных тестов, видео процесса тестирования и оценку соответствующих объектов тестирования относительно критериев выхода.

Использование различных подходов в тестировании интерфейса позволяет протестировать пользовательский интерфейс более тщательно, улучшить качество приложения и повысить удобство его использования.

II. ОБЗОР ТЕХНОЛОГИЙ ТЕСТИРОВАНИЯ ИНТЕРФЕЙСА ИГР

Методы разработки программного обеспечения не стоят на месте, то же можно сказать и об используемых для разработки и тестирования ПО инструментах и технологиях. Подбор качественного средства для тестирования интерфейса напрямую влияет на качество и скорость выполнения тестирования.

В настоящее время существует не малое количество инструментов тестирования интерфейсов мобильных приложений и игр. Наиболее популярные это:

- Appium;
- calabash;
- iOS UI testing от Apple;
- xUnit и TestNG.

Appium – наиболее известный сегодня инструмент. Он позволяет тестировать приложения практически вне зависимости от платфор-

мы, типа и версии системы. Appium поддерживает множество драйверов, что реализуется довольно интересным образом: Appium использует версию интерфейса WebDriver, известную всем по Selenium WebDriver.

Следующее весьма популярное средство тестирования — Calabash для Android и iOS. Инструмент разработан компанией Xamarin, однако с 2017 года его поддержкой занимается только сообщество. Calabash — довольно стабильный и быстрый инструмент, имеющий полезные функции для приведения приложения в нужное состояние и поддерживающий интеграцию с Cucumber.

iOS UI Testing — технология, представленная Apple на WWDC 2015. Как описывают её сами разработчики — это простой способ находить элементы пользовательского интерфейса, взаимодействовать с ними, а также проверять значения свойств этих элементов.

xUnit и TestNG — наиболее популярны фреймворки семейства xUnit. Они создавались как инструменты для unit-тестирования, и первым таким сервисом был JUnit. При этом они могут работать не только с модульными тестами, но и с любыми другими. Благодаря своей универсальности фреймворки xUnit используются повсеместно и доминируют в тестировании веб-приложений. JUnit работает только с Java, но сейчас есть реализации таких фреймворков практически под любой популярный язык программирования.

Все продукты в базовой комплектации без всяких особенностей бесплатны, любой разработчик может их добавить к своему проекту и пользоваться. iOS UI testing в отличие от своих конкурентов не является кроссплатформенным и подходит только для тестирования только игр на платформе iOS.

У Apple и Appium есть инструменты записи теста — во время взаимодействия с элементами интерфейса генерируется код теста, а у Calabash в базовой комплектации такого функционала нет. Однако она легко заменяется ана-

логами, например, Xamarin cloud. Также там предоставляется более 2000 девайсов, на которых можно выполнять тесты. Запись тестирования — не всегда очень хорошо. Когда тест не генерируется программистом, а автоматизируется, зачастую возникают ошибки, связанные с некорректным определением действий системой. Отсутствие такого инструмента может сыграть и в плюс.

Таким образом, каждый из описанных выше инструментов имеет свой уникальный набор возможностей, необходимый для решения постоянно меняющихся задач в области тестирования интерфейса мобильных игр. Благодаря таким инструментам появляется возможность осуществлять непрерывное тестирование и интеграцию, а также эффективно управлять процессом тестирования и получать отчет о проделанной работе. Данные инструменты отвечают всем требованиям, предъявляемым к тестированию мобильных и веб-приложений.

III. Выводы

Тестирование игр — сложный, но тем не менее интересный процесс, требующий качественной подготовки перед проведением. Тестирование интерфейса обладает рядом преимуществ — это увеличение надежности приложения, охватывание большей части пользовательских действий, проверка взаимодействия компонентов и сервисов между собой и повышение удобства использования игр.

Основные недостатки тестирования интерфейса игр — это большое количество времени, затраченное на описание всех компонентов и сервисов, а также то, что они малоприменимы к малым по объему приложениям. Недостатком же вышеперечисленных программ является отсутствие обучаемости системой на основании данных получаемых об игре, и как следствие, написание тестов занимает непозволительно много времени. Для решения этой проблемы, стоит обратить внимание на методы тестирования с использованием алгоритмов машинного обучения.

РЕНДЕРИНГ РЕАЛИСТИЧНЫХ ОБЪЕКТОВ НА ОСНОВЕ МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

Рак Т. А., Гриневич Я. Г.

Кафедра вычислительных методов и программирования,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: { tatianarak, ja.grinevich}@bsuir.by

Рассматривается базовая механика метода трассировки лучей – технологии построения реалистичных изображений, достоинства и недостатки.

ВВЕДЕНИЕ

Одним из методов рендеринга в компьютерной графике является метод «бросания лучей» (ray casting), при котором растровое изображение строится на основе замеров пересечения лучей с визуализируемой поверхностью в пространстве. Этот термин впервые использовался в компьютерной графике в 1982 году в работе Скотта Рота [1], который применил его для описания метода рендеринга CSG-моделей.

Первый алгоритм рейкастинга, используемый для рендеринга, был представлен Артуром Апелем в 1968 году [2]. В основе рейкастинга лежит идея генерировать лучи из точки наблюдения сцены, один луч на пиксель, и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта..

I. ПРИНЦИП МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

В основе метода трассировки лучей лежит идея генерировать лучи из точки наблюдения сцены, один луч на пиксель, и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта.

В реальной природе источник света испускает луч света, который, проходя через пространство, пересекает какую-либо преграду, которая прерывает распространение этого светового луча.

Алгоритм трассировки лучей (ray tracing) применяет рейкастинг для расчета первичных пересечений луча с объектами сцены и дополняет его генерацией дополнительных лучей для формирования световых бликов, теней, отражений, тем самым повышая уровень фотореалистичности изображения.

Для построения первичного луча и определения первых пересечений с объектами сцены вводятся понятия источника лучей и плоской области обзора. В основе модели лучевого эмиттера лежит механизм упрощенной камеры –

обскуры (рис. 1) с бесконечно малым отверстием, через которое свет проникает на область обзора.

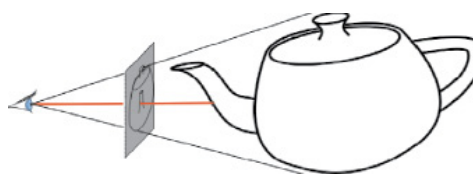


Рис. 1 – Принцип построения изображения

II. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ АЛГОРИТМА ТРАССИРОВКИ ЛУЧЕЙ

Определим плоскую область обзора как массив пикселей $\{pxm\}$, где n и m – количество пикселей в ширину и высоту соответственно, используя предварительно подготовленные структуры данных.

Установим начало координат в точку $O(x_0, y_0, z_0)$, центр камеры в точку $C(x_c, y_c, z_c)$. Зададим фокусный центр камеры точкой $F(x_f, y_f, z_f)$. Вектор $\vec{w} = \vec{FC}$ определяет направление камеры.

Для корректного определения направления распространения луча от позиции камеры к центру каждого пикселя плоской области обзора понадобится сформировать ортогональный базис $\{\vec{u}, \vec{v}, \vec{w}\}$ при помощи операций векторного произведения:

$$\vec{u} = \vec{w} \cdot \vec{y}_{0,1,0} \quad (1)$$

$$\vec{v} = \vec{u} \cdot \vec{w} \quad (2)$$

Направление первичного луча \vec{R} (рис. 2) определяется по следующим формулам:

$$\alpha = \tan\left(\frac{fov_x}{2}\right) \cdot \left(\frac{i - \left(\frac{n}{2}\right)}{\frac{n}{2}}\right) \quad (3)$$

$$\beta = \tan\left(\frac{fov_y}{2}\right) \cdot \left(\frac{\left(\frac{m}{2}\right) - j}{\frac{m}{2}}\right) \quad (4)$$

$$\vec{R} = \vec{OC} + \frac{\vec{w} + \alpha\vec{u} + \beta\vec{v}}{|\vec{w} + \alpha\vec{u} + \beta\vec{v}|} \quad (5)$$

где α, β – величины смещения луча по осям X, Y области обзора; i, j – целочисленные координаты пикселя, для которого генерируется луч;

fov_x, fov_y – горизонтальный и вертикальный углы обзора камеры.

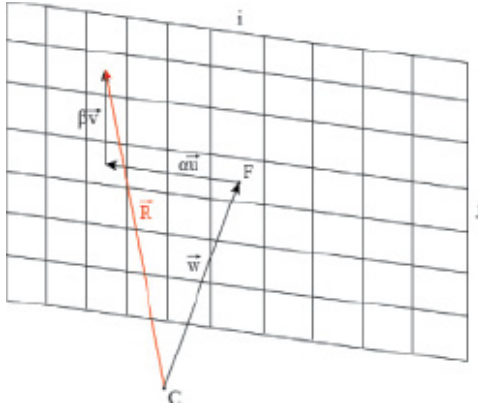


Рис. 2 – Вычисление направления распространения луча от позиции камеры к центру каждого пикселя координатной плоскости

Пусть $\vec{d} = \frac{\vec{w} + \alpha \vec{u} + \beta \vec{v}}{|\vec{w} + \alpha \vec{u} + \beta \vec{v}|}$.

Таким образом, можно записать векторное уравнение пучка лучей, исходящих из точки O с направляющими векторами \vec{d} :

$$\vec{R}_{i,j} = \vec{OC} + t\vec{d}, t \geq 0 \quad (6)$$

где t – расстояние от начала луча до любой точки на нем; \vec{d} – вектор направления распространения луча.

Рассмотрим алгоритмы определения пересечений с простейшим объектом – плоскостью.

Векторное уравнение плоскости, проходящей через фиксированную точку Q перпендикулярно вектору нормали \vec{n} :

$$\vec{n} \cdot \vec{QP} = 0, \quad (7)$$

где $P(x_p, y_p, z_p)$ – произвольная точка плоскости.

Решение системы уравнений (6) и (7):

$$\begin{cases} \vec{OC} + t\vec{d} \\ \vec{n} \cdot \vec{QP} = 0 \end{cases}$$

определяет значение параметра t , соответствующего точке пересечения луча с плоскостью:

$$t = \frac{\vec{n} \cdot \vec{OQ}}{\vec{n} \cdot \vec{d}}. \quad (8)$$

Если $t < 0$, то плоскость располагается позади камеры и луч ее не пересекает. Если $t \geq 0$, то точка пересечения находится на расстоянии $|\vec{OC} + t\vec{n}|$ от центра расположения лучевого эмиттера. Если $\vec{n} \cdot \vec{d} = 0$, то луч проходит параллельно плоскости и не пересекает ее.

III. ВЫВОД

К достоинствам метода трассировки лучей можно отнести следующие положения:

- возможность рендеринга гладких объектов без аппроксимации их полигональными поверхностями;
- вычислительная сложность метода слабо зависит от сложности сцены;
- отсечение невидимых поверхностей, перспектива и корректные изменения поля зрения являются логическим следствием алгоритма.

Серьезным недостатком метода является производительность. Метод растеризации и сканирования строк использует когерентность данных, чтобы распределить вычисления между пикселями. В то время как метод трассирования лучей каждый раз начинает процесс определения цвета пикселя заново, рассматривая каждый луч наблюдения в отдельности.

СПИСОК ЛИТЕРАТУРЫ

1. Roth, S. D. Ray casting for modeling solids // Computer Graphics and Image Processing. – 1982. – №18. – P. 109-144.
2. Appel, A. Some techniques for shading machine renderings of solids // AFIPS spring joint computer conference. IBM Research Center, Yorktown Heights, N.Y. – 1968. – P. 37-45.
3. Рак, Т. А., Реалистичный рендеринг на основе метода трассировки лучей / Т. А. Рак // 56-я научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» – 2020. – С. 147-148.

ВИЗУАЛИЗАТОР ТРЁХМЕРНЫХ МОДЕЛЕЙ ДЛЯ РЕАЛИЗАЦИИ ВИРТУАЛЬНЫХ ЛАБОРАТОРНЫХ И ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Савенко А. Г.

Кафедра информационных систем и технологий, Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: savenko@bsuir.by

В работе описан модуль визуализации 3D моделей для реализации виртуальных лабораторных и практических занятий. Алгоритмы визуализации моделей имеют оптимизации, позволяющие выполнять визуализацию на низкопроизводительных компьютерах, а сам визуализатор является кросс-платформенным.

ВВЕДЕНИЕ

Современная сфера образования претерпевает ряд изменений, обусловленных, в том числе, общими тенденциями цифровизации общества в целом. Виртуальная и дополненная реальности становятся более приемлемым способом доставки учебного контента [1]. Также такой подход имеет ряд существенных преимуществ, связанных со стремительно развивающейся дистанционной формы обучения [2]. Отдельным достоинством виртуализации обучающего контента является игровой подход при обучении, повышающий интерес и мотивацию к процессу обучения у молодых людей [3]. Эти преимущества можно реализовать при визуализации трёхмерных моделей лабораторных макетов, стендов и т.п. Разработанный модуль позволяет визуализировать модели, созданные практически в любых современных 3D редакторах, и встраивать их в систему управления обучением.

1. АЛГОРИТМЫ ВИЗУАЛИЗАЦИИ 3D МОДЕЛЕЙ

Поскольку современные графические процессоры рассчитаны на отрисовку треугольных фрагментов, 3D модель должна состоять из треугольников. Каждый треугольный фрагмент отдельно растеризуется графическим процессором на экран. В начале, в память видеокарты загружается информация о геометрии и материалах модели (текстуры альbedo, карты нормалей, металличности, шероховатости, фонового затенения и др.). Очистка экрана происходит при помощи стандартных алгоритмов, применяемых и в других визуализаторах. В вершинном шейдере происходит преобразование вершин моделей в нужный формат для отображения на экран монитора. Для быстрого преобразования модели в пространство NDC используются матричные преобразования. Для этого в вершинном шейдере каждую вершину модели нужно умножить на соответствующую матрицу итогового преобразования. Матрица итогового преобразования, в свою очередь, представляет собой произведе-

ние матриц вида, проецирования и локального преобразования. Матрица вида содержит информацию для отрисовки модели с нужной точки в пространстве (камеры) и имеет вид представленный на рисунке 1.

$$M_{view} = M_R M_T = \begin{bmatrix} l_x & l_y & l_z & 0 \\ u_x & u_y & u_z & 0 \\ f_x & f_y & f_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} l_x & l_y & l_z & -l_x x_c - l_y y_c - l_z z_c \\ u_x & u_y & u_z & -u_x x_c - u_y y_c - u_z z_c \\ f_x & f_y & f_z & -f_x x_c - f_y y_c - f_z z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

где l_x, l_y, l_z – нормированный вектор от камеры влево в соответствующих координатах;
 u_x, u_y, u_z – нормированный вектор от камеры вверх в соответствующих координатах;
 f_x, f_y, f_z – нормированный вектор от камеры к объекту в соответствующих координатах;
 x_c, y_c, z_c – координаты текущей позиции смотрящего (пространства глаза).

Рис. 1 – Матрица вида

Матрица проецирования позволяет визуализировать модель в перспективе и имеет вид представленный на рисунке 2.

$$\begin{bmatrix} \frac{1}{ar \cdot \tan(\frac{\alpha}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{\alpha}{2})} & 0 & 0 \\ 0 & 0 & \frac{-NearZ - FarZ}{NearZ - FarZ} & \frac{2 \cdot FarZ \cdot NearZ}{NearZ - FarZ} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

где ar – коэффициент между шириной и высотой прямоугольной области, на которую будет осуществляться проекция (коэффициент соотношение сторон);
 α – угол обзора глаза (камеры) по вертикали;
 $NearZ$ – позиция ближней Z-плоскости (позволяет обрезать объекты, находящиеся слишком близко к камере);
 $FarZ$ – позиция дальней Z-плоскости (позволяет обрезать объекты, находящиеся слишком далеко от камеры).

Рис. 2 – Матрица проецирования

Матрица локального преобразования хранит в себе данные о локальных трансформациях модели (угол вращения, перемещения в пространстве, масштабирования и т.д.) и, как правило, содержится в файле модели. Следующим основным этапом визуализации является фрагментный шейдер. Входными данными для него являются интерполированные данные из вершинного шейдера для каждого растеризуемого фрагмента и информация о материалах модели. Выходными данными является цвет фрагмента. Таким образом, в данном шейдере определяется цвет каждого фрагмента (пикселя) визуализированной модели. Для реалистичного отображения модели в данном шейдере просчитывается освещённость модели. Так как каждый фрагмент реагирует на свет по-разному, то необходимо принимать во внимание систему материалов (текстура альbedo, карты нормалей, метал-

личности, шероховатости, фонового затенения и кубические карты), приближенную к реальному миру и отражающую как каждый текстел нагруженный на модель текстуры реагирует на свет. Для имитации отражения и поглощения света от реального мира используется кубические карты, которые содержат вид вокруг какой-либо точки пространства модели. Однако для влияния пространства вокруг модели на какой-либо фрагмент необходимо вычислять интегралы по полусфере окружения пикселя в режиме реального времени, что является ресурсозатратной операцией (аналитический метод). Поэтому для ускорения учета влияния окружения на просчитываемый фрагмент заранее рассчитываются некоторые возможные варианты влияния. Суть такой оптимизации заключается в том, чтобы не высчитывать интеграл для каждого пикселя визуализации (для зеркального и диффузного освещения) в реальном времени, а брать уже готовые заранее подсчитанные значения для пикселей модели с определённым шагом. Данная оптимизация позволяет очень быстро рассчитывать фотореалистичное освещение даже на низкопроизводительных компьютерах (по сравнению с другими визуализаторами). Первая кубическая карта будет содержать в себе только диффузное освещение, где каждый вектор нормали (с шагом 1/255) будет содержать в себе диффузный свет в данном направлении. Остальные кубические карты будут содержать в себе зеркальное освещение, где каждый вектор нормали (с шагом 1/255) будет содержать в себе зеркальный свет (отражение) в данном направлении. Таких кубических карт зеркального освещения будет несколько, где каждая карта содержит значения для нужной шероховатости. Все промежуточные значения освещённости рассчитываются с помощью интерполяции. Конечная формула для расчета освещения представлена на рисунке 3.

$$L_o(p) = E_d(N) \cdot k_d + E_s(R, r) \cdot (k_s \cdot BF(NV, r)_r + BF(NV, r)_g),$$

где N – нормаль поверхности;
 V – вектор камеры;
 R – вектор отражения векторов V и N ;
 r – коэффициент шероховатости от 0 до 1;
 E_d – является выборкой из кубической текстуры диффузного освещения с координатами N ;
 E_s – является выборкой из кубической текстуры зеркального освещения с уровнем r и координатами N ;
 k_d – диффузный коэффициент материала;
 k_s – зеркальный коэффициент материала;
 BF – выборка из текстуры с координатами (NV, r) .

Рис. 3 – Формула расчета освещенности

В свою очередь, диффузный коэффициент материала определяется по формуле, представленной на рисунке 4.

$$k_d = albedo \cdot (1 - f_0) \cdot (1 - metallic),$$

где $albedo$, $metallic$ – параметры материала;
 f_0 – константа, принятая за 0,04.

Рис. 4 – Формула расчета диффузного коэффициента материала

Зеркальный коэффициент материала определяется по формуле, представленной на рисунке 5.

$$k_s = mix(f_0, baseColor, metallic),$$

где $baseColor$, $metallic$ – параметры материала;
 mix – функция интерполяции.

Рис. 5 – Формула расчета зеркального коэффициента материала

Следующим этапом расчёта освещения является просчёт прямого освещения от направленного источника света. Для этого используется карта теней. Для определения затенённости каждого пикселя фрагмента модели выполняется тест глубины (в два прохода), который и определяет, находится ли пиксель в тени или нет. Первый проход – это дополнительная визуализация с позиции источника света. Пиксели расположенные дальше от источника света и имеющие перекрывающие их пиксели не проходят тест глубины. Значение глубины ближайшего пикселя визуализируется в отдельный буфер глубины. При втором проходе сцена визуализируется с позиции смотрящего. Буфер глубины привязывается к фрагментному шейдеру. Для каждого пикселя получается соответствующее значение глубины из буфера. Если эти значения глубины равны, то его цвет находится стандартным алгоритмом. В противном случае делается выборка из карты глубины вокруг текущего пикселя и берётся их среднее значение. Для имитации влияния ярких участков изображения сначала размываются яркие участки изображения алгоритмом Гаусса, а затем смешиваются с основным изображением. На заключительном этапе визуализации применяются все рассчитанные эффекты для визуализируемой модели.

II. ЗАКЛЮЧЕНИЕ

Разработан модуль визуализации трёхмерных моделей, создаваемых практически в любых 3D редакторах. Полученный модуль имеет ряд преимуществ по сравнению с существующими аналогами, таких как: кросс-платформенность, невысокие системные требования к компьютеру за счёт оптимизированных алгоритмов вершинного и фрагментного шейдеров, возможность запуска в веб-браузере в режиме онлайн.

1. Савенко, А. Г. Виртуальная реальность, как способ получения и доставки учебного контента / А. Г. Савенко, Н. А. Кукалев, А. Г. Савенко // Высшее техническое образование : проблемы и пути развития : материалы IX Международной научно-методической конференции / редкол. : В. А. Богущ [и др.]. – Минск : БГУИР, 2018. – С. 394 – 397.
2. Савенко, А. Г. Преимущества и перспективы использования виртуальной и дополненной реальности в дистанционном образовательном процессе / А. Г. Савенко // Дистанционное обучение – образовательная среда XXI века : материалы X международной научно-методической конференции. – Минск : БГУИР, 2017. – С. 119.
3. Савенко, А. Г. Игровой подход в обучении программированию детей и подростков / А. Г. Савенко // Информационные технологии в технических, политических и социально-экономических системах : материалы Международной научно-технической конференции. – Минск : БНТУ, 2018. – С. 30.

ОБ ОДНОМ СВОЙСТВЕ ХАОТИЧЕСКИХ ПОТОКОВ С КРИВОЙ РАВНОВЕСИЯ

Цегельник В. В.

Кафедра высшей математики, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: tsegvv@bsuir.by

Исследован характер возможных подвижных особых точек решений динамических систем со скрытыми аттракторами и линией равновесия. Показано, что ни одна из четырех систем данного семейства не проходит тест Пенлеве. Проведен Пенлеве-анализ решений семейства из пяти динамических систем, обладающих хаотическим поведением и имеющих частные решения без подвижных особых точек. Доказано, что ни одна из систем указанного семейства не является системой Пенлеве-типа.

ВВЕДЕНИЕ

Общепризнанно, что математически простые системы дифференциальных уравнений могут проявлять хаос. С появлением быстродействующих компьютеров предоставляется возможным исследовать все пространство параметров этих систем с целью поиска параметров, которые приводят к некоторым желаемым характеристикам системы [1].

Так как одним из ключевых факторов в расчете колебаний нелинейных динамических систем является область притяжения, аттракторы можно разделить на самовозбуждающиеся и скрытые [2]. Самовозбуждающийся аттрактор имеет бассейн притяжения, который ассоциируется с неустойчивым равновесием. Скрытый аттрактор – это аттрактор, область притяжения которого не содержит окрестностей равновесия.

В работе [3], используя систематический компьютерный поиск, были найдены четыре простых хаотических потока с кубическими нелинейностями

$$\begin{cases} \dot{x} = z, \\ \dot{y} = -z(y^2 + xz), \\ \dot{z} = x^2 + y^2 - 1 + z(y^2 - z^2 + x^2), \end{cases} \quad (1)$$

$$\begin{cases} \dot{x} = -z, \\ \dot{y} = z(z^2 - 1), \\ \dot{z} = x^2 - y^2 - 1 + z(y^2 - z^2), \end{cases} \quad (2)$$

$$\begin{cases} \dot{x} = 0, 6z, \\ \dot{y} = z(0, 3y^2 + 0, 5xz), \\ \dot{z} = y^2 - 1 - xyz, \end{cases} \quad (3)$$

$$\dot{x} = -2z, \dot{y} = -z^3, \dot{z} = x^2 + y + z(z - xy), \quad (4)$$

для которых характерна необычная черта обладания кривой равновесия. Такие системы принадлежат к вновь представленному классу хаотических систем со скрытыми аттракторами, которые важны в инженерных исследованиях, поскольку они допускают неожиданные и потенциально катастрофические реакции на возмущения в такой конструкции, как мост или крыло самолета.

I. ПОСТАНОВКА ЗАДАЧИ

Целью работы является исследование характера возможных подвижных особых точек (т.е. точек, положение которых зависит от начальных условий) решений динамических систем (1)–(4), а также

$$\dot{x} = y, \dot{y} = z, \dot{z} = -x - 4y^2 + 11xz + 7yz, \quad (5)$$

$$\dot{x} = y, \dot{y} = z, \dot{z} = -z + 0, 7x^2 - y^2 + 0, 7xy + xy, \quad (6)$$

$$\dot{x} = y, \dot{y} = z, \dot{z} = -5x - 4y - y^2 + xz, \quad (7)$$

$$\begin{cases} \dot{x} = y, \\ \dot{y} = z, \\ \dot{z} = -x - y - z - y^2 - z^2 + xy + yz + 1, \end{cases} \quad (8)$$

$$\begin{cases} \dot{x} = y, \\ \dot{y} = z, \\ \dot{z} = -1, 01x - y - 1, 02z - y^2 + xy + 1 \end{cases} \quad (9)$$

с неизвестными функциями x, y, z в предложении, что независимая переменная t является комплексной.

Каждая из систем (5)–(9) имеет хаотическое поведение [4] и вместе с тем допускает частные решения без подвижных особых точек. А именно, системы (5)–(7) имеют решения

$$x = e^{-\tau}, y = -e^{-\tau}, z = e^{-\tau},$$

а системы (8)–(9) – решения

$$x = \varepsilon \sin \tau, y = \varepsilon \cos \tau, z = -\varepsilon \sin \tau,$$

где $\tau = t - t_0$ (t_0 – произвольная постоянная), $\varepsilon^2 = 1$.

II. АЛГОРИТМ

Для решения поставленной задачи использован тест Пенлеве [5], представляющий набор условий, необходимых для отсутствия у общего решения системы дифференциальных уравнений подвижных критических особых точек (свойство Пенлеве). Для анализа решений систем (5)–(9) использован подход, заключающийся в замене каждой из них эквивалентным уравнением третьего порядка и сравнением его с известными уравнениями, являющимися уравнениями Пенлеве-типа.

III. ВЫВОДЫ

Установлено, что ни одна из систем (1)–(4) не проходит тест Пенлеве. Показано, что ни одна из систем (5)–(9) не является системой Пенлеве-типа. Полученный результат согласуется с известной гипотезой [5], согласно которой выполнение для системы свойства Пенлеве с большой долей уверенности считается несовместимым с хаотичностью ее поведения.

С помощью теста Пенлеве в [6] был проведен Пенлеве-анализ решений систем дифференциальных уравнений

$$\dot{x} = xy, \dot{y} = xz, \dot{z} = x(-x + 1, 54y^2 - xz), \quad (10)$$

$$\dot{x} = xy, \dot{y} = x(-x + z), \dot{z} = x(3y^2 - xz), \quad (11)$$

$$\dot{x} = x(y^2 + 2xy), \dot{y} = -xz, \dot{z} = x(1 + xy), \quad (12)$$

$$\dot{x} = -yz, \dot{y} = x(x+z), \dot{z} = z(2y^2 + xz - 0, 35), \quad (13)$$

$$\begin{cases} \dot{x} = -0, 4xyz, \\ \dot{y} = xy(1 + z^2 - xy), \\ \dot{z} = xy(x^2 - xy), \end{cases} \quad (14)$$

$$\begin{cases} \dot{x} = xyz(y + 2yz), \\ \dot{y} = xyz(8z + y^2 + 7z^2), \\ \dot{z} = xyz(x^2 - y^2), \end{cases} \quad (15)$$

$$\begin{cases} \dot{x} = 0, 4y(1 - x^2 - y^2 - z^2), \\ \dot{y} = xz(1 - x^2 - y^2 - z^2), \\ \dot{z} = (1 - x^2 - y^2 - z^2)(-z - x^2 - 6yz), \end{cases} \quad (16)$$

$$\begin{cases} \dot{x} = (1 + y^2)(1 - x^2 - y^2 - z^2), \\ \dot{y} = (5x^2 - y)(1 - x^2 - y^2 - z^2), \\ \dot{z} = -xy(1 - x^2 - y^2 - z^2), \end{cases} \quad (17)$$

$$\begin{cases} \dot{x} = (y^2 - 5xy)(1 - x^2 - y^2), \\ \dot{y} = xz(1 - x^2 - y^2), \\ \dot{z} = (1 - 7y^2)(1 - x^2 - y^2), \end{cases} \quad (18)$$

$$\begin{cases} \dot{x} = (0, 1 - z^2)(1 + x^2 - y^2), \\ \dot{y} = xz(1 + x^2 - y^2), \\ \dot{z} = (y + xz)(1 + x^2 - y^2), \end{cases} \quad (19)$$

$$\begin{cases} \dot{x} = yz(z + x^2 + y^2), \\ \dot{y} = (x - xz)(z + x^2 + y^2), \\ \dot{z} = (x - 0, 6z^2)(z + x^2 + y^2), \end{cases} \quad (20)$$

$$\begin{cases} \dot{x} = yz(z + x^2 - y^2), \\ \dot{y} = -0, 1x(z + x^2 - y^2), \\ \dot{z} = (-z + 6y^2 + xz)(z + x^2 - y^2). \end{cases} \quad (21)$$

Системы (10)–(21) представляют новый класс хаотических систем со скрытыми аттракторами: системы с поверхностями равновесия. Они получены в [7] используя систематический компьютерный поиск. Доказано, что ни одна из систем (10)–(21) не проходит тест Пенлеве.

1. Sprott, J. C. *Elegant chaos: Algebraically simple chaotic flows* / J. C. Sprott. – World scientific: Singapore, 2010. – 304 p.
2. Леонов, Г. А., Кузнецов, Н. В. *Скрытые колебания в динамических системах: шестнадцатая проблема Гильберта, гипотезы Айзермана и Кальмана, скрытые аттракторы в контурах Чуа* / Г. А. Леонов, Н. В. Кузнецов // *Современная математика. Фундаментальные направления.* – 2012. – Т. 26. – С. 105–121.
3. Barati, K., Jafari, S., Sprott, J. C., Pham, V. -T. *Simple chaotic flows with a curve of equilibria* / K. Barati, S. Jafari, J. C. Sprott, V. -T. Pham // *International Journal of Bifurcation and Chaos.* – 2016. – Vol. 26, № 12. – P. 1630034 -1–6.
4. Faghani, Z., Nazarimehr, F., Jafari, S., Sprott, J. C. *Simple chaotic systems with specific analytical solutions* / Z. Faghani, F. Nazarimehr, S. Jafari, J. C. Sprott // *International Journal of Bifurcation and Chaos.* – 2019. – Vol. 29, № 9. – P. 1950116 -1–11.
5. Горизли, А. *Интегрируемость и регулярность / А. Горизли* // М. – Ижевск: НИЦ «Регулярная и хаотическая динамика»; Институт компьютерных исследований, 2006. – 316 с.
6. Цегельник, В. В. *Аналитические свойства решений семейства трехмерных автономных хаотических систем с поверхностями равновесия* / В. В. Цегельник // *Математическое моделирование и дифференциальные уравнения: материалы IV междунар. науч. конф., посвященной 95-летию со дня рождения чл.-кор. АН БССР, проф. Иванова Е. А. (Респ. Беларусь, Гродно, 17–19 дек. 2019 г.) / Ин-т математики НАН Беларуси. БГУ, ГрГУ им. Я. Купалы, редкол.: В. И. Корзюк (гл. ред.) и др. – Гродно: ГрГУ, 2019. – С. 99.*
7. Jafari, S., Sprott, J. C., Pham, V. -T., Volos, C., Li, C. *Simple chaotic 3D flows with surfaces of equilibria* / S. Jafari, J. C. Sprott, V. -T. Pham, C. Volos, C. Li // *Nonlin. Dynamics.* – 2016. – Vol. 86. – P. 1349–1358.

Секция «Системы обработки
информации»

Section «Systems of information
processing»

ОБЗОР МЕТОДОВ СТАТИЧЕСКОГО АНАЛИЗА ПРОГРАММНОГО КОДА

Азаренко А. В.

Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: lehaazarenko@mail.ru

Рассматривается возможность оптимизации и повышения защищенности приложений посредством статического анализа программного кода.

ВВЕДЕНИЕ

Статический анализ можно рассматривать как автоматизированный процесс обзора кода. Существенный недостаток методологии ручного обзора кода, это крайне высокая цена, поэтому компромиссным решением являются инструменты статического анализа кода, которые постоянно обрабатывают исходные тексты программ и выдают программисту рекомендации обратить повышенное внимание на определенные участки кода. Инструменты статического анализа кода используют комбинации методов статического анализа программного кода.

I. ОСНОВНЫЕ ЗАДАЧИ И ПРИЕМУЩЕСТВА

В целом задачи, решаемые с помощью методов статического анализа кода можно разделить на три категории:

1. Выявление ошибок в программном коде;
2. Рекомендации по оформлению программного кода. Некоторые платформы статического анализа позволяют проверять программный код на соответствие заданному стандарту оформления программного кода;
3. Подсчет метрик, позволяющих получить численное значение того или иного свойства программного обеспечения.

Главное преимущество статического анализ программного кода состоит в возможности существенного снижения стоимости устранения дефектов в программном обеспечении. Чем раньше ошибка выявлена, тем меньше стоимость ее исправления.

Применение методов статического анализа программного кода позволяет выявить большое число ошибок на этапе конструирования, или написания программного кода, что существенно снижает стоимость разработки всего проекта.

Статический анализ программного кода обладает рядом других преимуществ. Одним из них является полное покрытие кода, что позволяет находить дефекты и уязвимости в обработчиках редких ситуаций, ручное тестирование которых затруднено по тем или иным причинам. Статический анализ является независимым от среды использования и компилятора, что

в свою очередь позволяет обнаруживать ошибки, которые могут проявиться в процессе длительного использования ПО.

II. ОБЗОР МЕТОДОВ НА ПРИМЕРЕ ИНТЕРВАЛЬНОГО АНАЛИЗА

Интервальный анализ вычисляет для каждой целочисленной переменной верхнюю и нижнюю границы для ее возможных значений. Интервалы являются интересными результатами анализа, поскольку эти результаты могут использоваться для оптимизации и обнаружения ошибок, связанных с проверкой границ массива, числовых переполнений и представления целых чисел. Данный случай включает решётку бесконечной высоты, и мы должны использовать специальный метод для обеспечения сходимости к фиксированной точке [1].

Решетка, описывающая одно абстрактное значение, определяется следующим образом:

$$Intervals = lift(\{[l, h] \mid l, h \in N \wedge l \leq h\}), \quad (1)$$

где

$$N = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, +\infty\} \quad (2)$$

это множество целых чисел с бесконечным количеством конечных точек и порядком интервалов, определенных путем включения:

$$[l_1, h_1] \sqsubseteq [l_2, h_2] \Leftrightarrow l_2 \leq l_1 \wedge h_1 \leq h_2 \quad (3)$$

Эта решётка не имеет конечной высоты, поскольку она содержит, к примеру, следующую бесконечную цепь:

$$[0, 0] \sqsubseteq [0, 1] \sqsubseteq [0, 2] \sqsubseteq [0, 3] \sqsubseteq [0, 4] \sqsubseteq [0, 5] \dots \quad (4)$$

Это переносится на решетку абстрактных состояний

$$States = Vars \rightarrow Intervals \quad (5)$$

До указания правил ограничения необходимо определить функцию eval, которая выполняет абстрактную оценку выражений:

$$eval(\sigma, X) = \sigma(X) \quad (6)$$

$$eval(\sigma, I) = [I, I] \quad (7)$$

$$eval(\sigma, input) = [-\infty, +\infty] \quad (8)$$

$$\begin{aligned} & eval(\sigma, E_1 \text{ op } E_2) \\ = & \widehat{op}(eval(\sigma, E_1), eval(\sigma, E_2)) \end{aligned} \quad (9)$$

Абстрактные арифметические операторы определяются как:

$$\begin{aligned} \widehat{op}([l_1, h_1], [l_2, h_2]) = & \left[\min_{x \in [l_1, h_1], y \in [l_2, h_2]} x \right. \\ & \left. \text{op } y \quad \max_{x \in [l_1, h_1], y \in [l_2, h_2]} x \text{ op } y \right] \end{aligned} \quad (10)$$

Например:

$$\widehat{+}([1, 10], [-5, 7]) = [1 - 5, 10 + 7] = [-4, 17] \quad (11)$$

Функция JOIN является обычной для форвардного анализа:

$$JOIN(v) = \prod_{w \in pred(v)} [w] \quad (12)$$

Теперь мы можем указать правило ограничения для присвоений:

$$\begin{aligned} X = E : [v] = & JOIN(v)[X \\ \mapsto & eval(JOIN(v), E)] \end{aligned} \quad (13)$$

Для остальных узлов ограничение является тривиальным:

$$[v] = JOIN(v) \quad (14)$$

Теорема Клини о неподвижной точке утверждает следующее: всегда существует уникальное наименьшее приемлемое решение для такого анализа. Однако, так как интервальный анализ использует решетку с бесконечной высотой, теорема неподвижной точки не применима. В таком случае возникает вопрос: имеют ли ограничение интервального анализа наименьшее приемлемое решение для любой программы? Ответ положительный.

Один из способов определить тот факт, что наименьшая неподвижная точка существует, является более продвинутый вариант теоремы о неподвижной точке, который основывается на преобразовании итерационных последовательностей и выполнение без предположения о конечной высоте. Другим способом является использование другой теоремы о неподвижной точке, также известной, как теорема Тарского о неподвижной точке: в конечной решетке L каждая монотонная функция $f : L \rightarrow L$ имеет уникальную наименьшую неподвижную точку, полученную из $lfp(f) = \prod \{x \in L | f(x) \sqsubseteq x\}$.

Доказательство довольно просто. Допустим $D = \{x \in L | f(x) \sqsubseteq x\}$ и $d = \prod D$. Сначала покажем, что d это неподвижная точка f , то есть

$f(d) = d$. Предположим $x \in D$. Тогда $d \sqsubseteq x$, потому что d это нижняя граница D . Из монотонности f получаем $f(d) \sqsubseteq f(x)$ и $f(x) \sqsubseteq x$, так как $x \in D$. Таким образом, $f(d) \sqsubseteq x$, $f(d)$ также нижняя граница D . Так как d это наибольшая нижняя граница D , получаем $f(d) \sqsubseteq d$. Из монотонности f получаем $f(f(d)) \sqsubseteq f(d)$. Из этого следует, что $f(d) \in D$ и так как d является нижней границей D , получаем $d \sqsubseteq f(d)$. Из антисимметрии \sqsubseteq мы получаем $f(d) = d$. Чтобы убедиться в том, что d это уникальная наименьшая неподвижная точка f , предположим, что d' это какая-либо неподвижная точка f , то есть $f(d') = d'$. Тогда $d' \in D$ и, так как d это нижняя граница D , мы получаем $d \sqsubseteq d'$, а из антисимметрии \sqsubseteq следует тот факт, что наименьшая неподвижная точка уникальна.

Из-за того, что ограничение функций для интервального анализа монотонно, данная теорема свидетельствует о том, что наименьшая неподвижная точка существует. Таким образом, ограничения для интервального анализа всегда имеют четко определенное наиболее точное решение. Тем не менее, мы не можем вычислить наименьшую неподвижную точку, используя алгоритмы фиксации точки. Интервальный анализ решетки имеет бесконечную высоту, так что применение алгоритмов фиксации точки может никогда не заканчиваться: для решетки L_n последовательность приближений

$$f^i(\perp, \dots, \perp_n) \quad (15)$$

никогда не должна сходиться. Мощным инструментом для обхода подобного рода задачи является применение методов расширения и сужения.

III. Выводы

Метод интервального анализа используется для решения проблемы неопределенного потока мощности в централизованном кластере, так что в интервал включаются все решения, что, в свою очередь, позволяет покрыть все возможные варианты использования. Однако стоит отметить тот факт, что ни один из методов статического анализа программного кода не является ультимативным, поэтому выбор того или иного метода статического анализа программного кода зависит от большого числа факторов таких, как среда разработки и применения, архитектурные особенности продукта и т.д.

СПИСОК ЛИТЕРАТУРЫ

1. Андрес Меллер и Махаэль И. Шварцбах, Статический анализ программ / А. Меллер, М. И. Шварцбах // 2019. – С. 75-77.

АЛГОРИТМЫ СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКА ОБУЧЕНИЯ И ТРЕНИРОВКИ НЕЙРОННЫХ СЕТЕЙ

Белошедов Е. С., Гуринович А. Б., Гаруля Д. В., Архипенко Я. С.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: 2595118@mail.ru

В работе исследуется возможность повышения эффективности обучения нейронных сетей за счёт алгоритма оптимизации под названием стохастический градиентный спуск, показывающий необходимость изменять вес и смещение для минимизации потерь. Предложенный алгоритм позволяет ускорить процесс обучения и снизить количество корректировок параметров нейронной сети.

ВВЕДЕНИЕ

Нейронные сети способны решать широкий круг задач машинного обучения – прогнозирование временных рядов [1], распознавание речи [2], компьютерное зрение [3] и т. д. Актуальность проблемы обучения нейронных сетей связана с увеличением объемов данных, а так же с разнообразием архитектур сетей. Перед применением нейронной сети на практике, её необходимо обучить. Этот процесс является крайне требовательным к вычислительным мощностям процессора, а также видеокарты. Основная задача становится не просто обучение, а нахождение нейронной сети, наилучшим образом решающей поставленную прикладную задачу. Существует семейство методов нахождения такой сети, основанных на эмпирическом исследовании. Использование этих методов предполагает, что обучение является операцией решения задачи оптимизации структуры сети. Следовательно потребность в быстром обучении еще больше возрастает. Таким образом, для эффективного обучения нейронных сетей необходим новый алгоритм обучения.

I. АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ

Базовые компоненты нейронной сети – нейроны. Нейрон представляет собой единицу обработки информации в нейронной сети. Нейрон принимает входные данные, выполняет с ними определенные математические операции, а затем выводит результат (см. рис. 1).

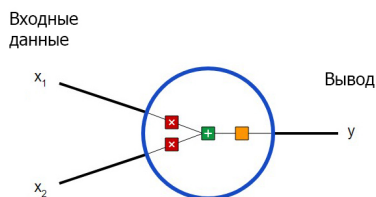


Рис. 1 – Нейрон с двумя входными данными

В первую очередь каждый вход умножается на вес: $x_{1,2} \rightarrow x_{1,2} * w_{1,2}$. Затем все взвешенные входы складываются вместе со смещением b : $x_1 * w_1 + x_2 * w_2 + b$. Затем сумма передается через функцию активации: $y = f(x_1 * w_1 + x_2 * w_2 + b)$. Функция активации используется для подключения несвязанных входных данных с выводом, у которого простая и предсказуемая форма. Как правило, в качестве используемой функцией активации берется функция сигмоида: $\sigma(x) = \frac{1}{1+e^{-x}}$ (см. рис. 2).

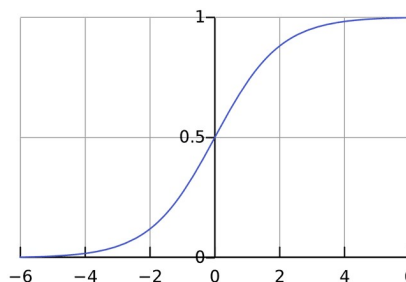


Рис. 2 – Функция сигмоида

II. МЕТОД СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКА

Персептрон – простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов. Нейронные сети часто обучаются стохастически, то есть на разных итерациях используются разные части данных. Это определяется, как минимум, двумя причинами: во-первых, наборы данных, используемые для обучения, часто очень большие, чтобы хранить их полностью в оперативной памяти и/или производить вычисления эффективно; во-вторых, оптимизируемая функция обычно невыпуклая. Таким образом, использование разных частей данных на каждой итерации может помочь от застревания модели в локальном минимуме. Кроме того, обучение нейронных сетей обычно производится с помощью градиентных методов первого порядка,

так как из-за большого количества параметров в нейронной сети невозможно эффективно применять методы более высоких порядков. Стандартным методом обучения нейронных сетей является метод стохастического градиентного спуска (SGD). Однако он может расходиться или сходиться очень медленно, если шаг обучения настроен недостаточно аккуратно. Поэтому существует много альтернативных методов, с целью ускорения сходимости обучения и избавить пользователя от необходимости тщательной настройки гиперпараметров [4]. Эти методы часто более эффективно вычисляют градиенты и адаптивно изменяют шаг обучения по итерациям. Суть данного алгоритма отражается в следующем уравнении: $\omega_1 \leftarrow \omega_1 - \eta \frac{\partial L}{\partial \omega_1}$, где η является константой, которая называется оценкой обучения, которая контролирует скорость обучения. Процесс тренировки нейронной сети на основе стохастического градиентного спуска будет выглядеть следующим образом:

1. Выбирается один пункт из набора данных;
2. Подсчитываются все частные производные потери по весу или смещению;
3. Применяется уравнение SGD, для обновления каждого веса и смещения;
4. Возврат к первому пункту.

У данного алгоритма можно выделить ряд преимуществ:

- Ошибка на каждом шаге считается быстро, веса меняются сразу же, что очень сильно ускоряет обучение. Обучение может сойтись ещё до того, как был выполнен единственный проход по всем тренировочным примерам. К тому же не надо хранить всю обучающую выборку в памяти;
- Стохастический градиентный спуск работает «более случайно», чем обычный, и поэтому можно надеяться, что он не остановится в маленьких локальных минимумах. Пакетный же спуск хорош для строго выпуклых функций, потому что уверенно стремится к минимуму глобальному или локальному;
- Подходит для онлайн-обучения, т. е. в случаях, когда обучающая выборка постоянно обновляется.

В данном методе присутствует недостаток — обновлять веса модели после каждого тренировочного примера может быть накладно, поэтому можно скрестить два этих варианта, получив мини-пакетный (mini-batch) спуск, который за раз обрабатывает, к примеру, 100 элементов, а не все или один. За счёт возможности распараллеливания это всё равно быстрее, чем в случае с пакетным спуском, а результат даёт даже лучше (см. рис. 3) [5].

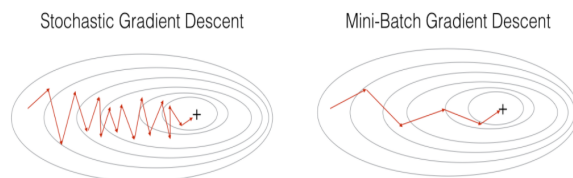


Рис. 3 – Сравнение методов градиентного спуска

Идеи иннерционных методов применяются для стохастического градиентного спуска и на практике часто дают прирост, в теории же обычно считается, что асимптотическая скорость сходимости не меняется из-за того, что основная погрешность в стохастическом градиентном спуске обусловлена дисперсией.

III. ЗАКЛЮЧЕНИЕ

При обучении нейросети типа «персептрон» требуется изменять весовые коэффициенты сети так, чтобы минимизировать среднюю ошибку на выходе нейронной сети при подаче на вход последовательности обучающих входных данных. Формально, чтобы сделать всего один шаг по методу градиентного спуска (сделать всего одно изменение параметров сети), необходимо подать на вход сети последовательно абсолютно весь набор обучающих данных, для каждого объекта обучающих данных вычислить ошибку и рассчитать необходимую коррекцию коэффициентов сети (но не делать эту коррекцию), и уже после подачи всех данных рассчитать сумму в корректировке каждого коэффициента сети (сумма градиентов) и произвести коррекцию коэффициентов «на один шаг». Показано, что при большом наборе обучающих данных алгоритм будет работать крайне медленно, поэтому на практике часто производят корректировку коэффициентов сети после каждого элемента обучения. Здесь значение градиента аппроксимируются градиентом функции стоимости, вычисленном только на одном элементе обучения показанного стохастического градиентного спуска является одной из форм стохастического приближения. Теория стохастических приближений даёт условия сходимости метода стохастического градиентного спуска.

IV. СПИСОК ЛИТЕРАТУРЫ

1. Kolarik, Thomas, and Gottfried Rudorfer. "Time series forecasting using neural networks." ACM Sigapl Apl Quote Quad. Vol. 25. No. 1. ACM, 1994.
2. Waibel, Alexander, et al. "Phoneme recognition using time-delay neural networks." Acoustics, Speech and Signal Processing, IEEE Transactions on 37.3 (1989): 328-339.
3. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012
4. Mnih, Volodymyr, Nicolas Heess, and Alex Graves. "Recurrent models of visual attention." Advances in Neural Information Processing Systems. 2014.
5. Stochastic Gradient Descent - Mini-batch and more. (2017, March 30). Adventures in Machine Learning.

ВЫБОР ОПТИМАЛЬНОЙ ОБЛАЧНОЙ ПЛАТФОРМЫ ДЛЯ ОРГАНИЗАЦИИ ХРАНИЛИЩА И ОБРАБОТКИ ДАННЫХ

Бессараб З. И.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: zekker6@gmail.com

Данная статья рассматривает проблемы выбора провайдера облачных услуг предоставляющего возможность размещения платформы для организации хранилища и центра обработки данных.

ВВЕДЕНИЕ

В современных информационных системах все чаще используются механизмы анализа и принятия решений на основе использования исторических данных. Например, для определения предпочтений пользователей системы построения рекомендаций создают профиль пользователя путём сопоставления действий пользователя с аналогичными действиями, произведенными другими пользователями. Такие подходы анализа данных требуют возможности оперативного анализа и хранения больших объемов данных.

Согласно отчету об исследовании, проведенному компанией IDC [1], объем данных, созданных в следующие три года, превысит общее количество информации созданной за 30 предыдущих лет. По прогнозу об объеме прироста количества новых данных ожидается ежегодное увеличение общего объема данных не менее чем на 20% в год.

I. ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ СОБСТВЕННЫХ ЦОД

Интенсивность роста объема данных приводит к тому что при использовании наиболее дешевых вариантов развертывания систем хранения и анализа данных – развертывания в собственных центрах обработки данных – существенная часть работ, связанных с обеспечением бесперебойной работы системы, сводится к наращиванию мощности системы. Недостаточная скорость увеличения мощности системы может вызывать: ухудшение качества обработки данных, понижение скорости ответов системы, технические сбои во время работы системы.

Наиболее популярным подходом, позволяющим решить проблему скорости масштабирования, является использование облачных платформ. Использование публичные облачных платформ позволяет производить автоматическое масштабирование ресурсов необходимых как для хранения, так и для обработки информации. В связи с этим возникает следующая проблема: необходимо определить облачную платформу, которая будет удовлетворять требованиям к производительности, возможности масшта-

бирования и стоимости решения до начала эксплуатации облачной системы.

II. СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ OLAP СИСТЕМ

Для получения этой информации производится нагрузочное тестирование целевой платформы при помощи использования бенчмарков. Одним из наиболее известных подходов [2] к бенчмаркингу является применение бенчмарка TPC-DS предназначенного для тестирования аналитических систем анализа данных. Однако не существует единого стандарта для анализа данных, полученных в результате проведения тестирования системы с использованием данного бенчмарка. Большинство существующих исследований [3–5] рассматривают лишь два показателя: стоимость платформы и время обработки данных. Данные показатели не позволяют провести ранжирование по другим показателям, например, количеству обращений к хранилищу данных при выполнении запроса.

III. АЛГОРИТМ ТЕСТИРОВАНИЯ ПЛАТФОРМ

Для создания более универсальной системы, позволяющей уменьшить сложность принятия решения по переносу рабочей нагрузки в облачную систему, необходимо разработать программное средство, которое на основании данных о тестировании облачных систем и существующей системы позволит определить облачную платформу, подходящую к заданным входным параметрам. Для создания такого продукта необходимо описать условия для проведения тестирования, произвести тестирование, создать продукт для анализа полученных данных и ранжированию сравниваемых систем по различным характеристикам.

Предлагается производить тестирование следующим методом: с помощью бенчмарка TPC-DS создаются тестовые наборы с различными объемами данных (например, три выборки с размерами 1гб, 10гб, 100гб). Затем определяется количество симулируемых одновременно активных клиентов (например, от 10 до 50 с шагом в 10 клиентов). Для каждой пары из наборов исход-

ных данных проводится следующий тест: в течение часа поочередно запускаются запросы из тестовой выборки TPC-DS. При этом в результате работы каждого запроса должны быть получены как минимум следующие данные:

1. Объем тестовой выборки
2. Количество активных пользователей
3. Название запроса
4. Время выполнения, секунды
5. Затраченные ресурсы CPU, секунды
6. IO, кб

Благодаря полученным данным возможно построить модели нахождения зависимости между различными метриками полученными в результате работы, а также возможно вычислить функцию зависимости затраченного количества ресурсов от объема датасета и загруженности платформы. Это позволяет определять необходимое количество ресурсов для работы системы без проведения дополнительных тестов.

IV. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Данные полученные в результате данного тестирования можно использовать для выбора оптимальной облачной платформы по различным критериям сравнения. На рисунке 1 показан график относительного взвешенного среднего времени отклика трёх платформ для набора данных размером 10 ГБ.

График показывает относительное, а не абсолютное время отклика, так как в данном случае мы исследовали степень влияния увеличения количества конкурирующих пользователей платформы на время отклика и не сравнивали значения абсолютного времени отклика среди рассмотренных платформ.

По показателю времени выполнения запроса в данном объеме данных видно что платформа Vantage показывает наименьшее время отклика.

ЗАКЛЮЧЕНИЕ

Проведя тестирование с использованием предложенного алгоритма тестирования были получены практические результаты в виде результатов тестирования. Эти данные могут быть использованы для дальнейшего исследования показателей производительности различных платформ, сравнения архитектурных подходов платформ позволяющих показывать лучшие показатели производительности.

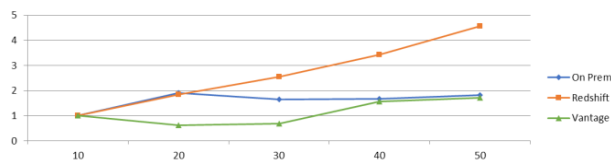


Рис. 1 – Зависимость времени отклика от количества активных пользователей

1. IDC's Global DataSphere Forecast Shows Continued Steady Growth in the Creation and Consumption of Data [Электронный ресурс]. – 2020. – Режим доступа: <https://www.idc.com/getdoc.jsp?containerId=prUS46286020> – Дата доступа: 02.10.2020.
2. Meikel Poess, Raghunath Othayoth Nambiar, and David Walrath. 2007. Why you should run TPC-DS: a workload analysis. In Proceedings of the 33rd international conference on Very large data bases (VLDB '07). VLDB Endowment, 1138–1149.
3. EDW performance comparison | Grid Dynamics Blog [Электронный ресурс]. – 2020. – Режим доступа: <https://blog.griddynamics.com/edw-performance-comparison/> – Дата доступа: 05.10.2020.
4. Evaluating modern data warehousing platforms with a performance per-dollar approach | West Monroe [Электронный ресурс]. – 2020. – Режим доступа: <https://www.westmonroepartners.com/perspectives/point-of-view/evaluating-modern-data-warehousing-platforms-with-a-performance-per-dollar-approach> – Дата доступа: 05.10.2020.
5. Cloud Data Warehouse Performance Testing | Gigaom [Электронный ресурс]. – 2020. – Режим доступа: <https://gigaom.com/report/cloud-data-warehouse-performance-testing/> – Дата доступа: 05.10.2020.

Таблица 1 – Пример агрегированных данных собранных в ходе выполнения тестирования

Объем тестовой выборки	Количество активных пользователей	Название запроса	Время выполнения, секунды	Количество выполнений	Затраченные ресурсы CPU, секунды	IO, кб
1	10	Query1	0.478	5767	2315	457549
1	10	Query2	0.795	4322	4.114	416.167
1	20	Query1	1.226	3799	2.410	102.335
1	20	Query2	1.964	3323	4.340	149.938
10	10	Query1	3.399	1165	18.108	297.692
10	10	Query2	5.032	1163	28.345	2380.264
10	20	Query1	7.497	1573	18.248	503.289
10	20	Query2	12.314	1570	28.914	670.933
100	10	Query1	34.907	123	178.049	71470.4
100	10	Query2	67.531	67	323.946	68077
100	20	Query1	58.548	107	176.095	14903.579
100	20	Query2	122.581	55	322.459	19861.106

МЕТОДЫ ПОЛУЧЕНИЯ И ОБРАБОТКИ ДАННЫХ В СИСТЕМАХ ПЕЧАТИ ДОКУМЕНТОВ НА ОСНОВЕ ПЕРЕМЕННЫХ ДАННЫХ

Бобков А. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: artez1997@gmail.com

В статье рассматривается система печати документов на основе переменных данных, а также методы получения и обработки этих данных. Анализируется предыдущий опыт обработки большого объема данных и улучшения, которые были внедрены на данный момент. Исследуются технические аспекты использования численных моделей.

ВВЕДЕНИЕ

В современном развивающемся мире количество используемых документов стремительно растет. Эффективное функционирование каждой организации, независимо от её профиля, находится в прямой зависимости от уровня оперативной обработки документации и информации, скорости взаимодействия между структурными подразделениями организации [1]. В современных условиях сокращение времени работы с данными возможно, прежде всего, из-за стремительного роста технологий. Поэтому автоматизация документооборота является одной из первоочередных задач современных организаций. Продуктами развития технологий являются различные системы, в частности, системы печати на основе переменных данных, которые были созданы для решения проблем, описанных выше.

I. СИСТЕМА VDP

Система VDP (Variable Data Printing) – это система, предназначенная для печати различного рода документов (контракты, финансовые отчеты и т.д.) на основе переменных данных [2]. Печать переменных данных объединяет набор основных элементов, которые являются общими для копий документа, с набором переменных элементов, которые изменяются от копии к копии. VDP использует технологию цифровой печати, которая связывает базы данных, которые содержат информацию для печати, с печатными устройствами. Данная коммуникация включает правила, которые определяют выбор информации из базы данных и размещения этой информации в документе.

Данная система включает в себя инструменты, которые позволяют пользователю вносить все необходимые изменения в документы без программирования, включая схемы данных, сопоставление данных из внешних файлов и сайтов, а также программирование бизнес-объектов и создание шаблонов. Система также включает в себя инструменты, которые обрабатывают любые данные, предоставленные пользователем, а

затем выдают результат в виде документов различных форматов (например, pdf, docx, pptx, xlsx и т.д.) [3]. Таким образом, решение состоит из двух частей:

1. Проектирование. Определение пользователем того, что должно быть напечатано и каким образом это должно быть отображено.
2. Обработка. Обработка определенных ранее данных и получение результирующего документа.

Схема работы системы представлена ниже (см. рис. 1).

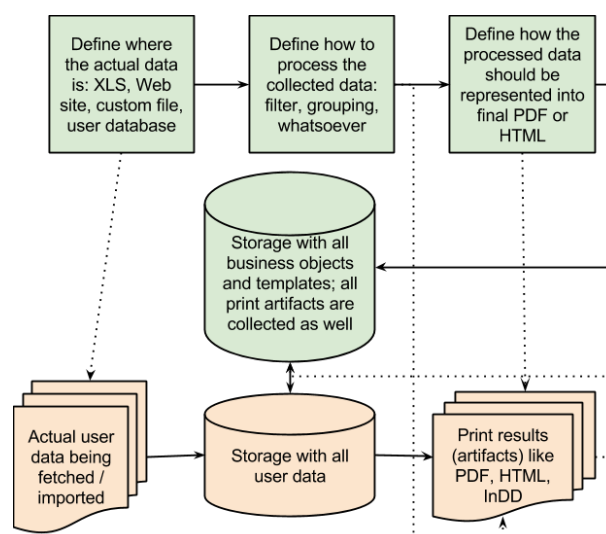


Рис. 1 – Схема работы системы печати VDP

На рисунке выше зеленый цвет означает первую стадию, то есть стадию проектирования будущего документа. Красный цвет означает стадию обработки, то есть стадию опеределения данных и получение финального документа.

Однако в такой системе очень важен вопрос получения и обработки данных для документов [4, 5]. Вопрос скорости импорта данных является ключевым. Таким образом, целью данного исследования является получение наилучшей производительности при получении и обработке дан-

ных, которые используются для печати документов.

II. ОБРАБОТКА ДАННЫХ

Исследование проводится на данных в объеме два гигабайта, которые содержатся в специальном файле формата .DAT. Затем данные файлы подвергаются обработке и приводятся в упорядоченный вид, после чего поступают в базу данных. Первоначальное время импорта данных в базу данных составляло 6 часов.

Далее было решено увеличить скорость, в первую очередь, за счет изменения метода анализа данных. Был разработан новый анализатор данных, который на основе содержимого файла получает множество xml-документов. Каждая строка файла представляет собой xml-документ, который затем подвергается обработке. Полученные в результате обработки данные представляют собой специально созданные структуры данных, которые отображают таблицы базы данных. Был проведен анализ обращения к полям таблиц базы данных и выявлены более оптимальные пути вставки и обновления новых данных [6]. Результаты работы нового алгоритма в сравнении с предыдущим представлены ниже (см. таблица 1).

Таблица 1 – Показатели производительности

Скорость импорта	Средняя	Максимальная	Минимальная
Алгоритм без использования нового метода обработки	21941.55 сек.	21231.6 сек.	22209.88 сек.
Новый алгоритм обработки данных	12843.16 сек.	13103.2 сек.	13935.77 сек.

Таким образом, в результате проведения исследования средняя скорость импорта данных в базу данных сократилась с 21231 секунд (около шести часов) до 13103 секунд (около трех с половиной часов), выигрыш по скорости импорта составил два с половиной часа, то есть обработка данных была ускорена почти в два раза.

На данный момент исследования в данном вопросе продолжаются, поэтому у этого анализа

есть потенциал для развития. В будущем планируется добавление импорта данных на основе различных форматов, например TXT, XLSX и т.д., а не только DAT. Также будет более подробно проанализировано количество обращений к определенным полям базы данных: получены дисперсия и математическое ожидание и другие статистические величины, составлена тепловая карта обращения к полям базы данных. В будущем массив обрабатываемых данных будет постоянно расширяться для обеспечения эффективной обработки всё большего и большего объема данных.

III. ЗАКЛЮЧЕНИЕ

Данная работа позволяет выявить наиболее оптимальные пути получения и обработки большого объема данных, которые в последствии используются для печати документов. В результате данного исследования был разработан новый анализатор данных, который за более быстрое время получает данные в форматированном виде, также были исследованы различные методы по оптимизации работы базы данных.

IV. СПИСОК ЛИТЕРАТУРЫ

1. Внедрение систем электронного документооборота: проблемы и решения [Электронный ресурс] – Режим доступа: <https://iteam.ru/publications/it/> – Дата доступа: 11.10.2020.
2. ABC's of VDP – A Variable Data Printing Basics Guide. [Electronic resource] – Mode of access: <http://help.efi.com/ref/vdp/en-us/VDP.pdf>. – Date of access: 11.10.2020.
3. Variable-Data Printing: Personalized Delivery of Information for the 21st Century [Electronic resource] – Mode of access: <https://web.archive.org/web/20111001035330/> – Date of access: 09.10.2020.
4. 3. Managing Variable Data Full-Color Printing at High Speeds [Electronic resource] /Hohensee, Reinhard. –InfoPrint Solutions, 2011. – Mode of access: <http://www.infoprintsolutionscompany.com/>. – Date of access: 12.10.2020.
5. 4. A New Generation of Variable Data Publishing Solutions.[Electronic resource] /Hohensee, Reinhard. – InfoPrint Solutions, 2011. – Mode of access: <http://www.infoprintsolutionscompany.com/>. – Date of access: 13.10.2020.
6. 5. Молинаро, К. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 672 с.

ИСПОЛЬЗОВАНИЕ ПРЕОБРАЗОВАНИЯ ГИЛЬБЕРТА-ХУАНГА ДЛЯ ЛОКАЛИЗАЦИИ КРАТКОВРЕМЕННЫХ ВОЗМУЩЕНИЙ ВИБРОСИГНАЛОВ

Бранцевич П. Ю., Лапицкая Н. В., Леванцевич В. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: branc@bsuir.edu.by, lapan@bsuir.by, lvn@bsuir.by

Рассматривается метод обнаружения кратковременных возмущений, возникающих в вибросигналах, полученных при виброконтроле машин роторного типа, который основан на разложении исходного вибросигнала на периодическую и шумоподобную составляющие. Исследуется возможность применения преобразования Гильберта-Хуанга для детального анализа кратковременных возмущений вибросигналов.

ВВЕДЕНИЕ

Важными показателями в технической диагностики механизмов роторного типа, главный рабочий элемент которых осуществляет вращательное движение, являются интенсивность и спектральный состав вибрации [1]. С целью обнаружения самых незначительных изменений в вибрационных сигналах необходимо осуществлять непрерывный контроль вибрации узлов для решения задач проактивного технического обслуживания таких механизмов.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ формы длинных реализаций вибрационных сигналов, отражающий интенсивность механических колебаний корпусных элементов механизмов роторного типа показал, что в большинстве случаев такие вибрационные сигналы содержат периодическую компоненту, и некоторую шумоподобную составляющую. При этом анализ шумоподобной составляющей на большом временном интервале позволяет выявить наличие кратковременных возмущений вибросигнала, которые плохо идентифицируются в спектре Фурье, но могут быть индикаторами зарождающихся дефектов [2]. Так при анализе вибросигнала с электродвигателя исследовательской установки (рис. 1, а) были выявлены кратковременные возмущения.

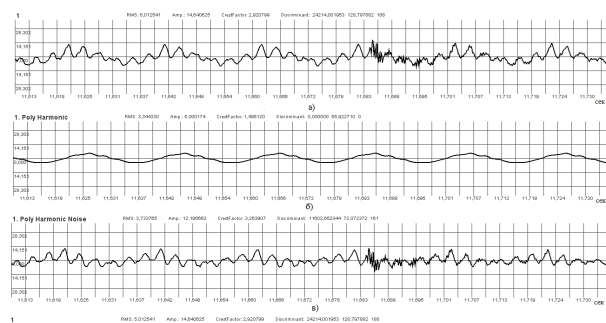


Рис. 1 – Разложение исходного сигнала (а) на периодическую (б) и шумоподобную (в) составляющие

Разложение вибросигнала на периодическую и шумоподобную составляющие (рис. 1, б, в) позволяет предварительно локализовать кратковременные возмущения [3]. Для повышения обнаруживающей способности возникающих возмущений и их более детального анализа, наиболее часто применяется вейвлет-преобразование шумоподобной составляющей [4]. На рис. 2 показан результат применения вейвлет-преобразования, к шумоподобной составляющей. В качестве преобразующего вейвлета был выбран вейвлет Морле (рис. 2, а).

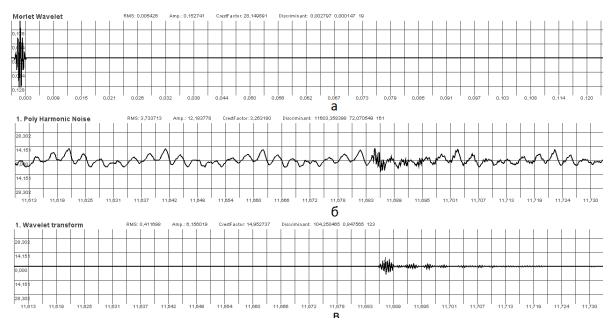


Рис. 2 – Результат преобразования (в) вейвлета Морле (а) и шумоподобной составляющей (б)

Как известно, основные сложности при использовании вейвлет-преобразования вызывает выбор типа базового вейвлета и центральной частоты его частотной характеристики [4].

II. РЕАЛИЗАЦИЯ

В качестве альтернативного способа локализации и анализа кратковременных возмущений к исходному шумоподобному сигналу было применено преобразование Гильберта - Хуанга, которое состоит из двух этапов: на первом этапе над сигналом выполняется эмпирическая модовая декомпозиция (ЭМД), в результате которой он разлагается на ряд компонент, которые называются эмпирическими модами (ЭМ), на втором этапе к полученному разложению применяется преобразование Гильберта [5]. Для реализации ЭМД на первом шаге необходимо опреде-

лить все максимальные и минимальные значения исследуемого вибросигнала $Y(t)$. На втором шаге по полученным минимальным и максимальным значениям строится верхняя и нижняя огибающие $Y_{max}(t)$ и $Y_{min}(t)$ исходного сигнала. При построения огибающей, на интервале между двумя экстремумами как правило, применяется интерполяция кубическими сплайнами. На третьем шаге вычисляется среднее значение огибающих по формуле:

$$m_1 = \frac{Y_{max} + Y_{min}}{2}$$

На четвертом шаге определяется первое приближение $h_{11}(t)$ к первой функции эмпирической моды $C_1(t)$:

$$h_1(t) = Y(t) - m_1(t)$$

Принимая вместо функции $Y(t)$ функцию $h_{11}(t)$ и повторяя шаги с первого по четвертый, находим второе приближение $h_{12}(t)$ к первой функции моды $C_1(t)$. Аналогичным способом находим и следующие приближения $h_{1k}(t)$ функции моды $C_1(t)$. Итерации продолжаются до тех пор, пока нормализованная квадратичная разность между двумя приближениями $h_{1k}(t)$ и $h_{1k-1}(t)$, не станет меньше некоторого предельного значения α :

$$\alpha \geq \sum_{t=0}^T \left(\frac{(h_{1k} - h_{1k-1})^2}{h_{1k-1}} \right)$$

Последнее значение $h_{1k}(t)$ представляет первую эмпирическую моду $C_1(t) = h_{1k}(t)$. Далее первая эмпирическая мода вычитается из исходного сигнала:

$$Y(t) - C_1(t) = r_1(t)$$

Для получения второй эмпирической моды $C_2(t)$ над остатком повторяются те же преобразования, что и для получения моды $C_1(t)$. Процесс получения n -го количества мод продолжается до тех пор, пока остаток $r_n(t)$ не станет монотонной функцией без экстремумов. Просуммировав полученные значения $C_i(t)$ и последний полученный остаток $r_n(t)$, который может быть трендом или постоянной составляющей сигнала, получим разложение $Y(t)$ в виде:

$$Y(t) = \sum_{i=1}^n C_i(t) + r_n$$

Процесс определения новых эмпирических мод может быть остановлен в следующих случаях: сумма модовых функций, полученная по выражению (1) отличается от исходного сигнала на величину, которая не превышает заранее установленного значения; остаточная функция становится монотонной функцией без экстремумов;

величина остатка незначительна в сравнении с исходным сигналом.

Результат разложения шумоподобного сигнала на нулевую и первую эмпирические моды приведен на рис.3.

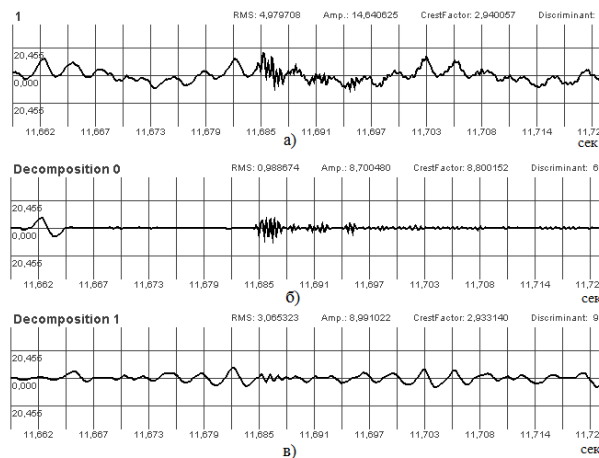


Рис. 3 – Результат разложения шумоподобного сигнала на эмпирические моды

Анализ нулевой моды (рис. 3, б) показывает, что её вид совпадает с результатом вейвлет-преобразования приведенного на рис. 2б.

III. ВЫВОДЫ

Преобразование Гильберта-Хуанга позволяет локализовать кратковременные возмущения вибрационных сигналов. В отличие от большинства классических подходов цифровой обработки сигналов, преобразование Гильберта-Хуанга не требует для своей реализации выбора функционального базиса разложения.

СПИСОК ЛИТЕРАТУРЫ

1. Неразрушающий контроль. Справочник. Том 7. Книга Вибродиагностика / Ф.Я. Балицкий и др. М.: Машиностроение, 2005. – 485 с.
2. Бранцевич П.Ю. Особенности применения алгоритмов цифровой обработки сигналов в системах вибрационного мониторинга и диагностики / П.Ю. Бранцевич, Е.В. Базылев, Е.К. Кулаковская // Доклады БГУИР – 2017. – №2 – С. 32–39.
3. Бранцевич, П. Ю. Способ анализа вибрационных сигналов при исследовании технического состояния механизмов / П.Ю. Бранцевич // Информационные технологии. Радиоэлектроника. Телекоммуникации (ITRT-2012) : сб. ст. II международной заочной научно-технической конференции. Ч. 1 / По-волжский гос. ун-т сервиса. – Тольятти : Изд-во ПВГУС, 2012. – с. 244 – 250
4. Бранцевич, П.Ю. Применение вейвлетов при анализе вибрационного состояния механизмов и агрегатов / П.Ю. Бранцевич // Актуальные вопросы машиностроения: сб. науч. тр. / Объедин. ин-т машиностроения НАН Беларуси; редкол.: С.Н. Поддубко [и др.]. – 2014. – Вып.3. – с. 220-228.
5. Feldman, M. Hilbert Transformation Application in Mechanical Vibration // M. Feldman // A John Wiley and sons, Ltd Publication – 2011. – 292p.

О МЕТОДЕ НАСТРОЙКИ УСТРОЙСТВА АЛАР КПАМ НА ОСНОВЕ РАВНОМЕРНЫХ ДАННЫХ

Бухаров Д. С.

Служба автоматизированных систем диспетчерского управления, Филиал АО «СО ЕЭС» Иркутское РДУ
Иркутск, Российская Федерация
E-mail: bukharovds@gmail.com

В работе представлен метод вычисления характеристики устройства АЛАР в форме трапеции. В качестве исходных данных используются годографы асинхронного режима. Метод основан на постепенном увеличении размеров характеристики с сохранением пропорций чувствительного и грубого органов характеристики. Основания трапеции вытягиваются до тех пор, пока автоматикой не будут зафиксированы все исходные годографы. Основной упор в процессе вычисления характеристики делается на соблюдение условий фиксации асинхронных режимов и чувствительности устройства АЛАР.

ВВЕДЕНИЕ

Настройка устройств автоматической ликвидации асинхронного режима (АЛАР) в большинстве случаев выполняется на основе неравномерных данных, полученных при расчете электрических режимов в различных программных комплексах. Такими неравномерными данными являются годографы асинхронного режима (АР) [1], в процессе вычисления которых при утяжелении режима увеличивается плотность точек. Ключевая проблема таких расчетов – отсутствие выдержки шага по времени.

В настоящей работе представлен метод настройки устройства АЛАР с трапециевидной характеристикой [2] на основе годографов АР с равномерным шагом по времени. Такие АР могут быть получены как расчетными средствами, так и зафиксированы регистраторами аварийных событий или устройствами синхронизированных векторных измерений.

Ранее в работе [3] были представлены алгоритмы настройки устройств АЛАР на основе неравномерных данных. Эти алгоритмы в процессе вычислений многократно изменяют форму и расположение трапеции (характеристики устройства АЛАР) с целью охвата максимального количества АР.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть имеется n годографов АР $G_i = \{R_{i,j}, X_{i,j}, t_{i,j} : j = \overline{1, m_i}\}$ ($i = \overline{1, n}$), содержащих по m_i замеров значений активного $R_{i,j}$ и реактивного $X_{i,j}$ сопротивлений и меток времени $t_{i,j}$, полученных с сохранением единого шага по времени ($t_{i,k+6} - t_{i,k+5} = t_{i,k+16} - t_{i,k+15} = \Delta t_i$).

Необходимо вычислить такую трапецию, при которой фиксируются все G_i ($i = \overline{1, n}$) и выполняются условия:

$$l_{SE/CE/h,top} > 0,1 \cdot l_{SE/CE/h},$$

$$l_{SE/CE/h,bot} > 0,1 \cdot l_{SE/CE/h},$$

$$l_{bas,top} \rightarrow \min,$$

$$l_{bas,bot} \rightarrow \min,$$

$$l_h \rightarrow \min,$$

$$T_{G_i} \geq T_{act},$$

где SE – чувствительный орган (ЧО); CE – грубый орган (ГО); h – высота трапеции; $l_{SE/CE/h}$ – длина ЧО/ГО/высоты трапеции; $l_{SE/CE/h,top}$ – длина отрезка, образованного пересечением ЧО/ГО/высоты трапеции и годографом АР, проходящего наиболее близко к верхнему основанию (рис. 1); $l_{SE/CE/h,bot}$ – длина отрезка, образованного пересечением ЧО/ГО/высоты трапеции и годографом АР, проходящего наиболее близко к нижнему основанию; $l_{bas,bot}$ – длина нижнего основания трапеции; $l_{bas,top}$ – длина верхнего основания трапеции; T_{act} – минимально необходимое время для фиксации АР; T_{G_i} – время нахождения i -ого годографа АР в ЧО характеристики.

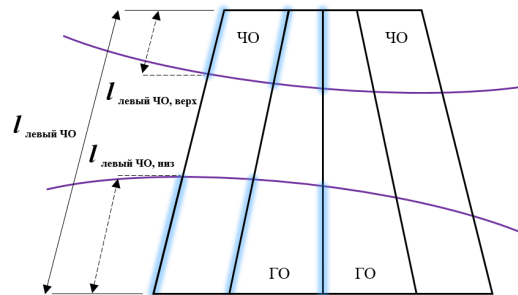


Рис. 1 – Условие чувствительности

Как отмечалось в работе [3] перед вычислением формы характеристики необходимо определиться с расположением органа направления мощности (ОНМ), на котором располагается высота трапеции. Размещается ОНМ любым удобным способом с соблюдением ключевого условия – пересечения ОНМ годографов АР в области наибольшей их плотности с целью получения трапеции минимальных размеров.

После определения расположения ОНМ выполняется построение трапеции согласно требованиям производителя устройств АЛАР [2].

II. МЕТОД ВЫЧИСЛЕНИЯ ХАРАКТЕРИСТИКИ

Для упрощения описания метода вычисления трапециевидной характеристики будем рассматривать пример с двумя годографами АР. Суть метода заключается в растягивании трапеции с соблюдением условия чувствительности и пропорциональности искомой фигуры (длина основания ГО равна половине длины основания ЧО), решая выписанную задачу (рис. 2).

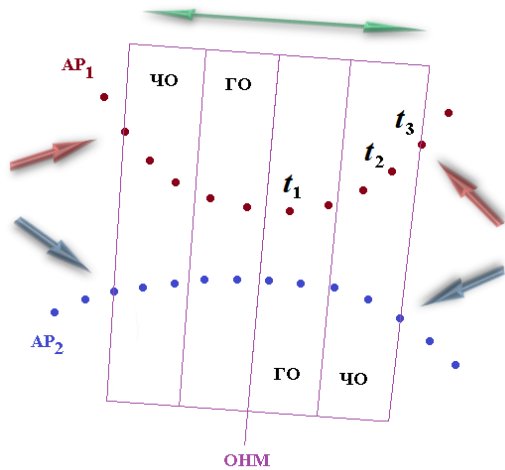


Рис. 2 – Построение характеристики

Метод состоит из следующих шагов:

ШАГ №1. Для каждого годографа АР определяются точки t_3 , которые взаимно наиболее удалены от ОНМ и обеспечивают выполнение условия $t_1 + t_2 \geq T_{act}$.

ШАГ №2. Через эти точки проводятся прямые – будущие боковины трапеции ЧО характеристики.

ШАГ №3. На построенных прямых откладываются необходимые отрезки с соблюдением условия чувствительности (рис. 1).

ШАГ №4. Строится трапеция с сохранением симметричности фигуры и пропорциональности ГО и ЧО.

ШАГ №5. Определяется количество АР, которые фиксируются устройством АЛАР.

ШАГ №6. Если фиксируются все АР, то вычисления останавливаются. В противном случае трапеция расширяется так, чтобы охватить еще по одной точке каждого годографа АР и выполняется шаг №2.

В случае трех и более годографов АР на шаге №2 прямая проводится через две геометрически наиболее удаленные от ОНМ точки.

III. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Описанный выше метод вычисления трапециевидной характеристики реализован программно [3]. Проведен вычислительный эксперимент на обезличенных данных: даны восемь годографов АР G_i ($i = \overline{1,8}$), мощность которых приблизительно равна и составляет

$m_i \simeq 1,4 \cdot 10^4$. $T_{act} = 20$ мс. Шаг по времени варьируется от 1 до 10 мс.

На рис. 3 отображена вычисленная характеристика и четыре годографа АР (G_1, G_4, G_5, G_6). Годографы представлены в усеченном виде: отображен только первый цикл. Все восемь АР фиксируются на первом цикле.

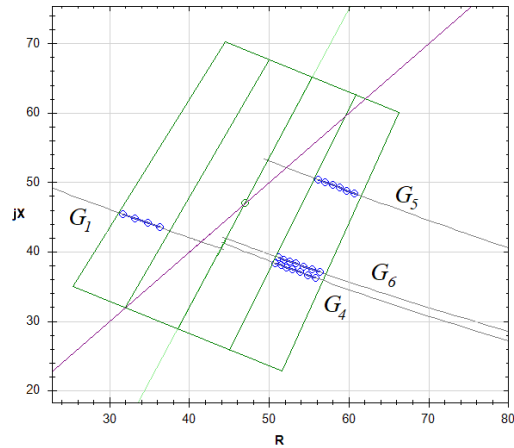


Рис. 3 – Годографы АР на первом цикле

Из рис. 3 видно, что количество точек, находящихся внутри ЧО характеристики, отличается. Шаг по времени равен $\Delta t_1 = 10$, $\Delta t_4 = 4$, $\Delta t_5 = 5$, $\Delta t_6 = 3$ мс. Наибольшее время нахождения АР в ЧО отмечается для G_1 , наименьшее время – G_6 .

Для рассматриваемого примера годограф АР G_6 является тем элементом, который ограничивает минимальный размер трапеции. Вычисление характеристики остановилось только после выполнения условия фиксации G_6 : ($T_{G_6} = 21$) > ($T_{act} = 20$).

ЗАКЛЮЧЕНИЕ

Представленный метод позволяет вычислить характеристику устройства АЛАР, удовлетворяющую условиям поставленной задачи. Полученная характеристика может использоваться специалистом по электрическим режимам как базовая при выборе уставок автоматики. При необходимости характеристика может быть скорректирована вручную в разработанном программном обеспечении [3] и проверена на фиксацию годографов АР.

СПИСОК ЛИТЕРАТУРЫ

1. Гоник, Я. Е., Иглицкий, Е. С. Автоматика ликвидации асинхронного режима / Я. Е. Гоник, Е. С. Иглицкий // М.: Энергоатомиздат, 1984. – 112 с.
2. Комплекс противоаварийной автоматики многофункциональный. 656455.206-01 РЭ. Новосибирск: Институт автоматизации энергетических систем, 2013. – 60 с.
3. Бухаров, Д. С. Об автоматизации настройки устройств автоматической ликвидации асинхронного режима с трапециевидной характеристикой / Д. С. Бухаров // Прикладная информатика. – 2017. – № 2. – С. 44–51.

ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ПРОГНОЗИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

Бык О. В.

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: bykalgerd@gmail.com

В данной статье автором рассматриваются основные классические и современные алгоритмы прогнозирования бизнес-процессов.

ВВЕДЕНИЕ

В настоящее время все более широкое применение находят алгоритмы для прогнозирования бизнес-процессов. С каждым годом конкуренция на рынках продуктов становится все более серьезной и все большее значение имеет правильное прогнозирование наперед различных тенденций и итогового результата. Как правило, алгоритмы прогнозирования достаточно тривиальны – осуществляется сбор исторических данных за прошлые периоды и на их основе составляются прогнозы на будущие периоды. Подобные методы являются действенными и успешными, однако ввиду постоянно усиливающейся конкуренции на рынке, в связи с чем такие алгоритмы прогнозирования становятся все более актуальными.

I. КЛАССИЧЕСКИЕ МЕТОДЫ ПРОГНОЗИРОВАНИЯ

Большинство методов, которые используются при решении задач искусственного интеллекта, базируются на регрессионных моделях и методах или используют их в качестве составных элементов. Важной их особенностью является развитый математический аппарат, с помощью которого можно оценивать качество построенных моделей, в частности, их точность и адекватность. Построение и исследование трех видов моделей: парной линейной регрессии (см. рис. 1), множественной линейной регрессии и некоторых типов нелинейной парной и множественной регрессии базируется на использовании метода наименьших квадратов. Вместе с тем, адекватное его применение требует выполнения определенного ряда предпосылок.

Метод наименьших квадратов (МНК), вне всяких сомнений, является тем классическим методом, с которого рационально начинать представление и обоснование методов прогнозирования. Он предназначен для оценки неизвестных величин по результатам измерения или экспериментов, содержащих случайные ошибки, и применяется для приближенного представления заданной функции другими (более простыми) функциями при обработке данных наблюдений. МНК предложен К. Гауссом и А. Лежандром.

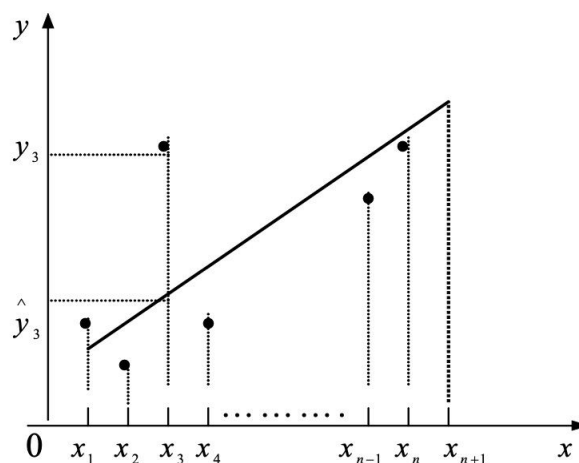


Рис. 1 – Парная регрессия

Для данного метода используются критерии оценки по минимуму суммы квадратов отклонений значений y , а также поиска минимума и максимума значений y . Применение первого из них направлено на уменьшение влияния отдельных «выбросов», а второго – приводит к более равномерному приближению во всех точках. Иное объяснение состоит в том, что они отвечают наблюдениям в условиях помех с разными статистическими свойствами. Если значения исходной характеристики определяются с точностью до нормально распределенного случайного слагаемого, то для оценивания коэффициентов используется первый функционал. Его применение оправдано также в условиях, если ошибка измерений распределена по закону Лапласа. Если слагаемое распределено нормально в некотором интервале, то для оценки параметров регрессии необходимо использовать второй функционал [1].

II. НЕЙРОСЕТЕВЫЕ МЕТОДЫ И ИСПОЛЬЗОВАНИЕ ИХ ДЛЯ ПРОГНОЗИРОВАНИЯ

В последние несколько лет мы наблюдаем взрыв интереса к нейронным сетям которые успешно применяются в самых различных областях – бизнесе, медицине, технике, геологии, физике. Нейронные сети вошли в практику везде, где нужно решать задачи прогнозирования, классификации или управления. Такой впечатляющий успех определяется несколькими причинами. Во-первых, нейронные сети – исключи-

тельно мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости. В частности, нейронные сети нелинейны по своей природе. На протяжении многих лет линейное моделирование было основным методом моделирования в большинстве областей, поскольку для него хорошо разработаны процедуры оптимизации. В задачах, где линейная аппроксимация неудовлетворительна (а таких достаточно много), линейные модели работают плохо. Кроме того, нейронные сети справляются с проблемой размерности, которая не позволяет моделировать линейные зависимости в случае большого числа переменных. Во-вторых, нейронные сети учатся на примерах. Пользователь нейронной сети подбирает представительные данные, а затем запускает алгоритм обучения, который автоматически воспринимает структуру данных [2].

Для прогнозирования бизнес-процессов популярно использование сетей встречного распространения (СВР). Данные сети предназначены для начального быстрого моделирования (см. рис. 2). Автор СВР Роберт Хехтнильсен.

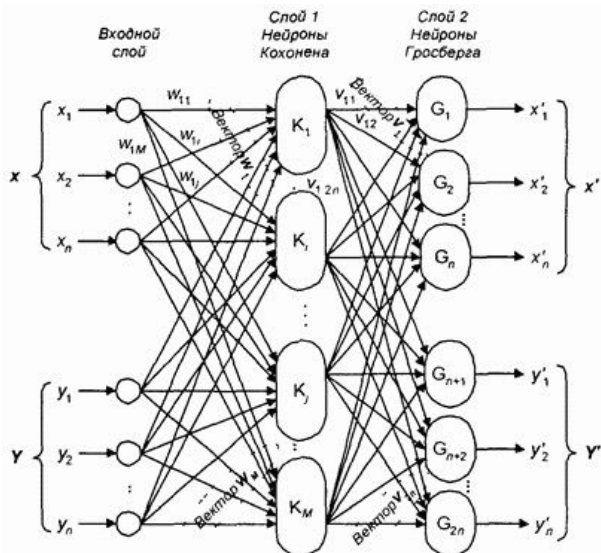


Рис. 2 – Сети встречного распространения

СВР работает с векторами, значениями которых являются непрерывные величины или двоичные, то есть такие, которые состоят из нулей и единиц. В результате обучения входные векторы ассоциируются с векторами на выходе сети и, если сеть обучена, подача входных образов приводит к получению результирующих образов.

СВР работает в двух режимах: обучения и использования. В первом случае на входы подаем и вектор X , и вектор Y , корректируем весовые коэффициенты; во втором режиме на вход уже обученной сети подаем X или Y , а на выходе получаем значения и X , и Y . Рассмотрим аспекты алгоритма функционирования СВР. Пер-

вое: нормируем каждый элемент таблицы. Далее генерируем случайным образом весовые коэффициенты и обязательно их нормируем, тем самым, сокращая процесс обучения. Здесь существует несколько нюансов, о них можно прочитать, например, в известной книге Ф. Уоссермана, а также частично ниже. После подаем на вход сети строку матрицы Z и рассчитываем скалярные произведения с векторами весовых коэффициентов, которые связаны со всеми нейронами слоя Кохонена. Среди всех скалярных произведений выбираем произведение с максимальным значением и настраиваем вес соответствующего нейрона. Выход нейрона, который выиграл, равен единице, все остальные – нулю. Значения выходного вектора слоя Кохонена подаются на слой нейронов Гроссберга. В каждом нейроне слоя Гроссберга обычным способом рассчитывается активация, корректируем все веса слоя Гроссберга и выводим результаты.

Применение такого алгоритма обучения является проблематичным, если значительное количество обучающих шаблонов образует совокупность незначительных размеров в области обучения. В таком случае результат работы классического алгоритма будет правильным и необходимо применять предварительную обработку данных [3].

Применение нейронных сетей для прогнозирования является перспективным в случае существования сложных нелинейных зависимостей со значениями факторов, на которые не наложено ни одного ограничения. Вместе с тем, необходимо заметить, что получение высокоточных результатов возможно только при наличии квалифицированных специалистов, поскольку необходимым условием эффективного функционирования нейросети является формализация задачи, предварительная подготовка данных и обеспечение избежания исключительных ситуаций [4].

III. ЗАКЛЮЧЕНИЕ

Были рассмотрены и проанализированы основные алгоритмы прогнозирования бизнес-процессов, начиная от алгоритмов, созданных на первых этапах, и заканчивая наиболее актуальными и перспективными алгоритмами нейронных сетей для построения прогнозов. Оценена актуальность использования нейронных сетей для прогнозирования бизнес-процессов.

1. Снытюк В. В. Прогнозирование. Модели, методы, алгоритмы / В. В. Снытюк – Киев: Национальный Университет имени Тараса Шевченко, 2008.
2. Назаров А. В., Лоскутов А. И. Нейросетевые алгоритмы прогнозирования и оптимизации систем / Назаров А. В., Лоскутов А. – Москва, 2017.
3. Хайкин С. Нейронные сети / Хайкин С. – Издательский дом "Вильямс 2006.
4. Бугорский В. Н. Использование нейронных сетей в работе трейдера / Бугорский В. Н. – Москва, 2011.

МЕТОДЫ ОПТИМИЗАЦИИ НА ОСНОВЕ ГРАДИЕНТНОГО СПУСКА С МОМЕНТОМ В ЗАДАЧАХ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Гаруля Д. В., Навроцкий А. А., Белошедов Е. С.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: dimagarul58@gmail.com

В данной работе исследуется возможность повышения эффективности обучения нейронных сетей за счёт модификации алгоритма градиентного спуска моментом и его вариацией под названием момент Нестерова, которые позволяют проводить физическую аналогию процессу оптимизации, как «скатывание» частицы в рельефе. Рассмотренные алгоритмы позволяют не только ускорить скорость обучения нейронных сетей но и уменьшить эффект их переобучения.

ВВЕДЕНИЕ

В настоящее время алгоритмы машинного обучения (АМО) достигли значительно уровня. АМО способны решать широкий круг задач – прогнозирование временных рядов, распознавание речи, компьютерное зрение и т. д. Характерной чертой АМО является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Обучение таких алгоритмов сводится к минимизации функции потерь. Основным методом оптимизации функций, применяемых в задачах машинного обучения, является метод градиентного спуска (ГС) [1]. Главным недостатком этого метода при использовании в обучении нейронных сетей (НС) является резкое падение скорости обучения на участках незначительного изменения функции ошибок («плато»), при этом теоретически для НС ГС всегда находит минимум функции. Таким образом, одной из важнейших задач для развития АМО является поиск наиболее эффективно метода оптимизации функций.

I. ПРИНЦИП МАШИННОГО ОБУЧЕНИЯ

Два ключевых компонента в контексте задач классификации документов:

- Параметризованная функция, сопоставляющая необработанный текст с меткой класса;
- Функция потерь, которая измеряет качество определенного набора параметров на основе того, насколько хорошо совпадают полученные оценки с заданными метками обучающих данных;

Дана непрерывная функция $J(\theta)$, определяет итеративный алгоритм оптимизации. Один из самых простых способов оптимизации - это градиентный спуск, который при заданных начальных параметрах, локально изменяет параметр θ на итерации t в сторону уменьшения значения J . $J(\theta_t + v_t) = J(\theta_t) + v_t J'(\theta_t) + O(v_t^2)$. Для маленькой скорости обучения $\alpha_t > 0$, заданная $v_t = -\alpha_t J'(\theta_t)$ уменьшает значение J . ПО-

этому итерация градиентного спуска сводится к $\theta_{t+1} = \theta_t + v_t$. Для выпуклой функции J градиентный спуск сходится к оптимальному значению. Хотя градиентный спуск весьма популярен, альтернативные методы, такие как момент или ускоренный градиент Нестерова (Nesterov's Accelerated Gradient (NAG)) может привести к значительно более быстрой сходимости к оптимальному значению [2].

II. ГРАДИЕНТНЫЙ СПУСК С МОМЕНТОМ

Градиентный спуск с моментом (Momentum Gradient Descent), как правило, обеспечивает более быструю сходимость в глубоких нейронных сетях. Для этого метода можно провести физическую аналогию. Функцию потерь можно представить как некоторый рельеф, при этом значение функции потерь соответствует значению некоторой высоты, то есть потенциальной энергии. Тогда процесс оптимизации мы можем представить как процесс «скатывания» частицы в этом ландшафте. Поскольку сила, действующая на частицу, связана с градиентом потенциальной энергии (то есть $F = -\nabla U F = -\nabla U$), сила, воспринимаемая частицей, является в точности (отрицательным) градиентом функции потерь. Более того, $F = ma F = ma$, поэтому (отрицательный) градиент в этом представлении пропорционален ускорению частицы. Обратите внимание, что это отличается от обновления стохастического градиентного спуска, показанного выше, где градиент напрямую определяется положением. Вместо этого физический вид предлагает обновление, в котором градиент напрямую влияет только на скорость $v = mu * v - learning_rate * dx$, которая в свою очередь, влияет на положение $x = x + v$. Сравнительный график представлен на рисунке 1. Водится переменная v , которая инициализируется нулем и дополнительный гиперпараметр mu . Гиперпараметр mu , при оптимизации, называется моментом, его физический смысл больше соответствует коэффициенту трения. Типовыми значениями гиперпарамет-

ра mu являются 0.5, 0.9, 0.95, 0.99. Фактически, эта переменная уменьшает скорость и кинетическую энергию системы, иначе частица никогда не остановится перед возвышением ландшафта. С обновлением гиперпараметра mu вектор параметров будет наращивать скорость в любом направлении, имеющем постоянный градиент.

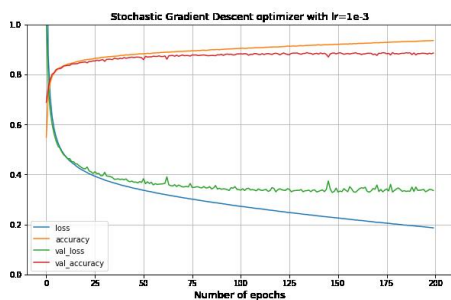


Рис. 1 – Потери и точность модели с набором данных Fashion MNIST

III. МОМЕНТ НЕСТЕРОВА

Nesterov Momentum - это немного другая версия набирающего популярность в последнее время метода Momentum Gradient Descent. В теории он обладает более высокой скоростью сходимости для выпуклых функций, а на практике он также работает немного лучше, чем стандартный момент. Основная идея момента Нестерова заключается в том, что когда текущий вектор параметров находится в некоторой позиции x , то, глядя на обновление момента по формуле представленной выше ($v = mu * v - learning_rate * dx$ (интегрированная скорость)). Только один член момента (т.е. игнорирование второго члена с градиентом) изменяет параметр vector с помощью $mu * v$. Следовательно, если мы собираемся вычислить градиент, мы можем рассматривать приблизительное «следующее» положение $x + mu * v$ как «опережающий взгляд» - это точка в непосредственной близости от того места, где мы скоро окажемся. Следовательно, имеет смысл вычислять градиент в точке $x + mu * v$ вместо «старой» позиции x [3]. Вместо того, чтобы оценивать градиент в текущей позиции (красный кружок), мы знаем, что наш импульс вот-вот приведет нас к кончику зеленой стрелки. Поэтому с помощью импульса Нестерова мы вместо этого оцениваем градиент в этой «прогнозируемой» позиции (см. рис. 2).

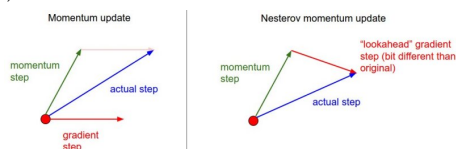


Рис. 2 – Принципы методов градиентного спуска с моментом и момента Нестерова

То есть в несколько неудобных обозначениях мы хотели бы сделать следующее:

- $x_ahead = x + mu * v$ оценить dx_ahead (градиент на x_ahead вместо x);
- $v = mu * v - learning_rate * dx_ahead$;
- $x = x + v$;

Однако на практике люди предпочитают выражать момент Нестерова так, чтобы он выглядело как можно более похожим на исходный метод стохастического градиентного спуска или на метод градиентного спуска с моментом. Этого можно достичь, используя переменную $transform$ $x_ahead = x + mu * v$, а затем выражая момент Нестерова в терминах x_ahead вместо x . То есть вектор параметров, который мы фактически сохраняем, всегда является опережающей версией. Уравнения в терминах x_ahead (но переименовывая его обратно в x) становятся:

- $v_prev = v$
- $v = mu * v - learning_rate * dx$ обновление скорости остается прежним
- $x = x - mu * v_prev + (1 + mu) * v$ обновление позиции изменяет форму

ЗАКЛЮЧЕНИЕ

Метод градиентного спуска, применяемый в обучении нейронных сетей, гарантирует сходимость к оптимальному значению при бесконечном количестве итераций. Однако на практике скорость обучения при использовании градиентного спуска оказывается невелика из-за эффекта «плато». Модификация градиентного спуска с использованием момента позволяет уменьшить этот эффект позволяя преодолевать участки с малым градиентом за меньшее время за счет накопленной «инерции». Вычисление градиента с некоторым опережением позволяет увеличить эффективность метода градиентного спуска с моментом.

СПИСОК ЛИТЕРАТУРЫ

1. Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157-166, 1994
2. Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. "On the importance of initialization and momentum in deep learning." Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML13, p. 1139, 2013
3. Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady, volume 27, p. 372-376, 1983

ONE VERSION OF THE GROUP RESOLUTION PRINCIPLE FOR DISCRETE OPTIMIZATION

German J. O.

Computer systems and networks department, The Belarusian State university of informatics and radioelectronics
Minsk, Republic of belarus

One modification of the group resolution principle (grp) to find a minimum-weighted covering set of 0,1-matrix is presented. The modification makes it possible to use grp in the same way both for non-weighted case of a minimum-size covering problem and for the weighted case. This enables one to use the single algorithm for the generalized covering problem. It is also shown how to modify the existing approach to avoid matrix sizes growing when adding new group resolvents at the iterations of the algorithm. The suggested technique is characterized by quite good computational complexity estimation showing the polynomial complexity of the method. Summarizing all this up, one can conclude that the method may find use for different optimization problems widely presented in data and text mining and the other areas.

INTRODUCTION

One of the interesting applied problems is finding a minimum-weighted covering set of the 0,1-matrix. This problem is widely presented in the different applied areas using data mining, text mining and the other optimization techniques. It may be used to solve logical inference problems, including that one known as satisfiability of the given system of disjuncts. Moreover, grp usage can be extended to the first order logic of predicates basing on the same ideas as outlined below. In [1,2], were described the theoretical backgrounds of the group resolution principle and its use for solving a 0,1-matrix covering problem with minimum-size (minimum-weighted) set of rows. Comparing the techniques, used for these problems, one can conclude that they are similar, though not identical. The difference lies in the way of resolvents making. A resolvent represents a new and unique 0,1-column generated in a special way and added to the current 0,1-matrix. It is proved that sooner or later the totally zero resolvent-column would be generated what points to the finishing of the searching procedure. The best covering set found till this moment defines the solution to the initial problem. In this paper, we show two main results. The first result consists in applying grp formulation to the weighted case of the 0,1-matrix covering problem, while the second results gives a possibility to limit the number of the column-resolvents added at the iterations of the algorithm. The necessary complexity estimations show the polynomial efficiency in average. The rest of this short report contains the details. We avoid considering illustrations for the interests of brevity and paper sizes restrictions. Instead, the narration was made as clear as possible.

I. EXPLANATION OF THE GRP

Let us remind the basic principles of grp [1]. The grp-based method uses a 0,1-matrix B with $n > 0$ nonempty rows and $m > 0$ nonempty columns.

One says that row i covers column j if i contains in j a unit. It is required to find a minimum-size set CV of rows such that each column of the initial matrix B is covered by at least one row from CV . The grp-based algorithm performs a finite number of iterations. At each iteration, the algorithm uses some heuristic to find a current covering set P . We use a kind of a greedy algorithm to find P . This greedy algorithm seeks the undeleted column c with minimum number of units and then selects an undeleted row r which covers c and contains maximum number of units among all the rows covering c . The column c is called a syndromic for the covering set P . This syndromic columns corresponds to the row r and the row r is included in P . Then all the columns of the current matrix B , covered by the row r , are (temporarily) being deleted. Also, all the rows covering the syndromic column c (including r) are being deleted as well. The iteration then continues to complete building the covering set P . Evidently, this process stops when there would be no undeleted columns in matrix B . By this, the current covering set P , consisting of the rows r_1, r_2, \dots, r_z , is found. Then the group resolvent is being formed according to grp. To form group resolvent, one should build a new matrix on the syndromic columns corresponding to the rows r_1, r_2, \dots, r_z . The matrix built on syndromic columns is called a syndromic matrix. Let us designate it by Sdr_p . It is used to generate a new unique column called a group resolvent res . The rule to generate res is the following one: res contains unit in the row i if and only if row i has two or more units in syndromic matrix Sdr_p . Otherwise, res contains zero in row i . Then, all temporarily deleted rows and columns of the matrix B are restored and res is added to B . Provided, that res contains at least one unit, a new iteration starts. The total finiteness of the computational process directly follows from the uniqueness of res what is proved in [1]. As for weighted case of the covering problem, the grp formulation is somewhat different. To form a syndromic matrix at the current

iteration, one seeks for a current covering set P as explained before. However, the group resolvent is being formed in a different way. For the weighted case of the covering problem, each row is assigned an integer non-negative weight w_i . The final solution should deliver a minimum value to $\sum w_i$ where index i defines the row number from the optimal covering set. Let as previously, Sdr_P stand for the syndromic matrix found at the current iteration for the covering set P . Let $w(P)$ stand for the weight of the covering set P . Divide all the rows into two subsets: S_1 and S_2 . Subset S_1 consists of the rows which contain in Sdr_P no more than one unit. Subset S_2 consists of the other rows (with more than one unit in each of them). For each column c_j in Sdr_P define a value of v_j corresponding to the minimal weight of the row from S_1 which covers c_j . Define the current low boundary Low_P of P the value $Low_P = \sum v_j$. Then, if $Low_P < w(P)$ one needs to move some row(s) from S_1 to S_2 to provide $Low_P \geq w(P)$ and form a group resolvent with units in the rows from S_2 . Otherwise, provided S_2 is empty, the process terminates with the answer, corresponding to the best covering set found. The details may be found in [2]. Now we show how to unify the Grp formulations for the weighted and non-weighted cases of the covering problem.

II. GRP UNIFICATION

The unified version of grp is obtained as follows. At each iteration, one seeks for the column with minimum number of units (this is a syndromic column c as earlier) and selects a row which covers it and has minimal weight among all the rows covering c . To be applicable to the non-weighted case of the problem, the conditional weights are introduced computed as $1/n_i$, where n_i stands for the number of units in the row i . So, the more units contains the row, the less is its weight. Now, let syndromic matrix Sdr_P be defined corresponding to the current covering set P . Evidently, there is no need to define the sets S_1 and S_2 . One can directly use the grp, formulated for the non-weighted case of the problem: the rule to generate res is the following one – res contains unit in the row i if and only if row i has two or more units in syndromic matrix Sdr_P . Otherwise, res contains zero in row i . We need to prove correctness of this rule. It should be clear that each column j in Sdr_P has value of v_j corresponding to the row with minimal weight from those covering column j due to heuristic used to form current covering set P . By this, $Low_P \geq w(P)$ and it is impossible to improve the solution, provided there are no rows with two or more units in Sdr_P . This formulation, evidently, does not require to move some rows from S_1 to S_2 and saves time to make the algorithm faster than in [2]. This, moreover, is not the single possible enhancement. The second

one is connected to restriction on the sizes of the 0,1-matrix B which are permanently growing in [1,2]. Consider, how one can do this. The idea is to use the previously added column-resolvents in order to overwrite them with new resolvents [3]. The common rule is as follows: if at the current iteration some previous group resolvent was not included into Sdr_P then it may be overwritten by a new generated resolvent without loss of solution. Provide the following reasoning. Let a cover P_i was found at iteration i by sequential including rows r_1, r_2, \dots, r_z . Suppose that new iteration $i + 1$ entirely repeats the previous iteration i . This means that the same syndromic columns and the same rows r_1, r_2, \dots, r_z are selected in the same order including some additional new row(s). At the moment of including row r_i into P_{i+1} (r_z is the last one in P_i) matrix B cannot be entirely destroyed, otherwise one gets $P_i = P_{i+1}$, which is impossible according to grp theoretical properties [1,2]. This means that at least one column c should remain undeleted and c is not covered by any one of rows r_1, r_2, \dots, r_z . But column c must be at this moment totally zero as all the rows having «1s» in c will be deleted (because the same syndromic columns are selected for rows r_1, r_2, \dots, r_z at the iterations i and $i + 1$). Evidently, this is impossible and enables one to come to one of the next conclusions: either P_{i+1} has less than z rows or at one of the steps $1, 2, \dots, z$ when forming cover P_{i+1} in the selected syndromic column there would be smaller amount of units in comparison to the syndromic column selected at the same step while forming cover P_i . The last remark concerns computation complexity of the method. Clearly, as the suggested technique is based on grp for non-weighted case of the minimum-size covering problem, the estimations should preserve the same order. According to [1], grp converges to a solution approximately for a number of steps estimated as $O(nmp/(1-p))$ where p denotes density of units in matrix B , n stands for the number of rows, and m stands for the number of columns. The expressions points to polynomial complexity for the value of p not close to 1 or 0. In conclusion, let us note that the unified version of grp is more natural for programming aims, clear for understanding and may be successfully used in the university courses dealing with discrete optimization.

1. Geman, O.V. Statistically optimal algorithm for the minimum-size covering problem / O. German, V. Naydenko // Economics and mathematical methods. Moscow. 1993. vol. 29, №4, p.p. 662–667 (in Russian).
2. German, Yu. An efficient in average algorithm for the covering problem with application to fuzzy classification and fuzzy inference / Yu. O. German et al. // Proceedings of BSUIR Scientific journal. (Minsk, Belarus). 2009, №7, p.p. 93–99 (in Russian).
3. German, O. Expert systems / O. V. German // Minsk, BSUIR, 2008. – p. 90 (in Russian).

ГЕНЕРАТОР ЭКВИВАЛЕНТНОГО ТЕКСТА НА БАЗЕ LSTM СЕТИ

Гуменников Е. Д., Мурашко И. А.

Кафедра информационных технологий, Гомельский государственный университет имени П. О. Сухого

Гомель, Республика Беларусь

E-mail: guma178@gstu.by

Данная работа посвящена методике применения нейронных сетей LSTM архитектуры для генерации текстов семантически идентичных предоставленным в качестве входных. Здесь рассматриваются особенности реализации подходящей нейронной сети, а также затрагивается методика сбора данных для обучения такой нейронной сети.

ВВЕДЕНИЕ

Цель данной работы состоит в адаптации нейросетевых методик автоматической генерации текстов, для решения задачи генерации текстов эквивалентных предоставленным в качестве входных данных. Система способная качественно решать подобную задачу может найти применение в таких отраслях как, разработка чатботов, голосовых интерфейсов, ну и конечно для автоматизации работы «копирайтеров».

Решение данной задачи на базе простой рекуррентной нейронной сети имеет вероятно окажется неэффективным. Это обусловлено тем что такая архитектура не имеет механизма «памяти», что накладывает серьезные ограничения на способность такой системы оценивать контекст встречающихся в тексте слов. Однако LSTM сеть обладает такой памятью.

По этой причине, в качестве базовой архитектуры нейронной сети, способной решить поставленную задачу, рассматривается LSTM сеть. Ранее сети данной архитектуры показывали наилучший результат в обработке последовательностей и текстов в частности.

I. МЕТОДИКА ПРИМЕНЕНИЯ LSTM СЕТИ В КОНТЕКСТЕ ЗАДАЧИ ГЕНЕРАЦИИ ЭКВИВАЛЕНТНЫХ ТЕКСТОВ

LSTM является наилучшей архитектурой для решения поставленной задачи. Так как эти ИНС обладают долгосрочной памятью они способны учитывать контекст всего исходного текста и генерировать материалы наиболее близкие исходным. Данная архитектура нейронных сетей, другими словами, способна обрабатывать всю последовательность элементов, рассматривая каждый элемент как часть последовательности, то есть каждое слово рассматривается как часть текста, как исходного, так и генерируемого. Учитывать контекст генерируемого текста также важно, как и исходного, так как новые элементы в генерируемой последовательности также являются частью текста.

LSTM типа многие ко многим отлично подходят для решения задач NLP, так как исходные

и выходные данные являются последовательностями неопределенной длины.

LSTM сети не требуют каких-либо особых модификаций для их успешного применения в контексте задачи генерации текста эквивалентного заданному.

Для решения поставленной задачи целесообразно применить подход "seq2seq". Исходный оригинальный текст подается на вход LSTM сети, носящей название кодировщик. Выход этой сети является состоянием ячейки, полученным при обработке последнего элемента исходного текста. Это состояние подается в качестве входных данных второй рекуррентной сети, которую называют декодировщиком, ее предназначение состоит в генерации последующего слова эквивалента. Ошибки декодировщика передаются кодировщику через состояние ячейки. Данный вектор в описанной модели называется вектором промежуточного представления. Промежуточное представление используется в популярных моделях предназначенных для решения задач автоматического перевода и, как правило, представляют граф представления интерпретирующий входной текст предназначенный для перевода. Система перевода генерирует выходной текст на основе этой промежуточной структуры. Подобная модель может быть применена и для решения задачи генерации эквивалентных текстов[3].

Генерация эквивалента из оригинального текста начинается с того, что первое слово оригинала подается в качестве входных данных в сеть, где оно вместе с информацией о предыдущих итерациях генерации следует к сигмоидальному слою, который называется «фильтром слоя забывания». Он определяет то, какую часть информации, содержащейся в состоянии ячейки следует «забыть». Так можно окинуть семантическую информацию, которая вероятнее всего более не понадобится при генерации эквивалента. Математически этот шаг описан в как:

$$f_t = \sigma(W_t(h_{t-1}, x_t) + b_f)$$

где W_t – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_f – величина смещения.

Далее необходимо по текущему состоянию ячейки и новоприбывшему слову оригинала определить то, какую часть новой информации следует добавить к состоянию ячейки, то есть «запомнить». Этот этап состоит из двух шагов. Во-первых, сперва сигмоидальный слой, называемый «слоем входного фильтра» определяет то, какие значения в состоянии ячейки следует обновить, затем тангенс слой строит вектор значений кандидатов на добавление к состоянию ячейки. Так сеть может запомнить ключевые слова и семантическую информацию исходного и генерируемого текста, для ее последующего употребления при генерации. Математическая интерпретация описанных процессов:

$$i_t = \sigma(W_i(h_{t-1}, x_t) + b_i)$$

где W_i – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_i – величина смещения.

$$C'_t = \tanh(W_C(h_{t-1}, x_t) + b_c)$$

где W_C – веса нейронных связей;

h_{t-1} – предыдущее слово сгенерированного эквивалента;

x_t – очередное слово оригинала;

b_C – величина смещения.

Обновление состояния ячейки таким образом происходит по следующему сценарию:

- старое состояние ячейки умножается на f_t , таким образом из состояния ячейки удаляется ненужная более информация;
- к состоянию ячейки прибавляется произведение C'_t и i_t .

Таким образом в состояние ячейки поступает новая информация о сгенерированном и об оригинальном тексте. Математически это можно записать в виде:

$$C_t = f_t C_{t-1} + i_t C'_t$$

где C_t – обновленное состояние ячейки;

f_t – коэффициент забывания;

C_{t-1} – предыдущее состояние ячейки;

i_t – вектор входного фильтра;

C'_t – вектор кандидатов на добавление к состоянию ячейки.

Далее необходимо сгенерировать новое слово эквивалентного текста. Для этого анализируется текущее состояние ячейки. Сперва применяется сигмоидальный слой, определяющий какую информацию из состояния ячейки необходимо применять. Затем состояние ячейки обрабатывается тангенс слоем, возвращающим вектор значений, величина которых находится в пределах от -1 до 1. Затем оба вектора перемножаются таким образом получается вектор значений кодирующий продолжение генерируемого текста. Затем результат отправляется в следующую итерацию, где будет обработано следующее слово оригинала. Также отправляется и новое состояние ячейки, таким образом формируется долгосрочная память о генерации и оригинальном тексте, что позволяет добиться наилучших результатов.

II. СБОР ДАННЫХ ДЛЯ ОБУЧЕНИЯ LSTM СЕТИ-ГЕНЕРАТОРА ЭКВИВАЛЕНТОВ

Для обучения подобной LSTM сети необходим огромный объем обучающих данных. Каждый элемент обучающей выборки представляет из себя пару оригинал-эквивалент.

Собрать такие данные удобно с помощью нескольких переводов зарубежной литературы выполненных разными авторами.

Предполагается выбирать фрагменты из двух экземпляров текстов содержащие идентичную семантическую нагрузку. Один фрагмент станет оригиналом второй будет примером эквивалента.

Полностью автоматизировать процесс сбора данных не представляется возможным, так как авторы довольно своеобразно форматируют текст.

1. Барто, Э. Обучение с подкреплением / Э. Барто // ДМК. – 2020. – С. 216–232.
2. Розанов, А. К. Быстрый алгоритм анализа словоформ естественного языка с трехуровневой моделью словаря начальных форм / А. К. Розанов. // РГРУ. – 2016. – С. 14.
3. Хайкин, С. Нейронные сети. Полный курс / С. Хайкин. // Вильямс. – 2006. – С. 912–987.
4. Тарик, Р. Создаем нейронную сеть / Р. Тарик. // Вильямс. – 2018. – С. 12–67.

ОСОБЕННОСТИ АРХИТЕКТУРНОЙ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ. ИНТЕГРИРОВАННЫЕ КОНВЕЙЕРНЫЕ ВЫЧИСЛИТЕЛИ

Жилияк Н. А.

Кафедра информационных систем и технологий, Белорусский государственный технологический университет

Минск, Республика Беларусь

E-mail: gznadya@gmail.com

Построение вычислительных систем реального времени в большинстве случаев связано с решением задачи обеспечения высокой производительности. Попытка решить проблему повышением быстродействия элементной базы приводит к пределу, обусловленному конечным значением скорости света. В данной связи наиболее перспективным направлением поиска приемлемых решений указанной задачи является поиск нетрадиционных архитектурных решений. Одним из таких решений, предложенных автором настоящей статьи, является подход, основанный на модернизации известного принципа организации вычислительного процесса — конвейеризации.

ВВЕДЕНИЕ

Одной из основных проблем, возникающих при проектировании вычислительных систем реального времени является достижимость требуемого быстродействия (производительности). Для последовательного выполнения алгоритма из I-операции необходим ресурс времени, который определяется как сумма произведений продолжительности выполнения i-й операции и такта или время цикла, которому пропорциональны моменты начала или завершения любых действий в системе. Повышение быстродействия элементной базы или, что то же самое, уменьшение значения такта имеет свой предел, ограниченный скоростью света.

Поэтому для решения названной проблемы более перспективными являются пути поиска архитектурной организации вычислительной системы (ВС), связанные, в первую очередь, с совмещением операций. Два основных подхода в этом направлении — конвейеризация и параллелизм.

I. ОСНОВНАЯ ЧАСТЬ

Отметим некоторые особенности объектов, о которых пойдет речь в этой работе. Во-первых, уточним термин «Вычислительная система», под которым будем понимать совокупность взаимосвязанных и взаимодействующих функциональных и/или конструктивно законченных вычислительных модулей, предназначенных для решения определенного ряда задач или одной конкретной задачи по обработке, передаче или сохранению информации. В дальнейшем указанные модули будем называть функциональными устройствами, под которыми будем понимать компоненты самых разных иерархических уровней, от элементарных функциональных элементов или групп элементов, выполняющих простейшие функции в составе БИС, ПЛМ или БМК, до процессоров, сопроцессоров или целых ЭВМ.

Такая точка зрения на термин «вычислительная система» позволяет распространить его на технические средства от микро- до макроуровней, или от БИС или их основных фрагментов до мощных комплексов, систем и сетей. Во-вторых, разговор будет вестись про вычислительные системы реального времени, это значит, про вычислительные системы, работающие в режиме реального времени — режиме обработки данных, при котором взаимодействие ВС с внешними по отношению к ним процессам осуществляется в моменты, определяемые скоростью протекания этих процессов. Отметим, что требование реализации режима реального времени обуславливает при проектировании таких ВС в качестве основной решение задачи обеспечения временных соотношений между их компонентами как основы построения таких ВС, поэтому доминирующую роль в предложенных методах занимает концепция организации вычислительного процесса в реальном времени, в то время как вопросы пространственного построения ВС РВ могут стать темой дальнейших исследований в этой отрасли [1, 3]. Третья особенность ВС, на проектирование которых направлены предложенные автором методы, также связана с требованиями режима реального времени, согласно с которыми должна обеспечиваться обработка данных сразу после их поступления, а также выдача результатов или управляющей информации в требуемые интервалы времени параллельно для разных внешних объектов. В этом случае будем ориентироваться на параллельно-конвейерные вычислительные архитектуры — архитектуры с множественными потоками данных, обработка которых по параллельным ветвям подразумевает конвейеризацию.

По способу организации вычислительного процесса функциональные устройства разделим на простые и конвейерные. Функциональные устройства называется простым, если время выполнения на нем любой операции определено

априори и никакая дальнейшая операция не может начать выполняться раньше момента завершения предыдущей операции. Функциональные устройства называются конвейерными, если время выполнения на нем любой операции также определено заранее, но дальнейшая операция может начать выполняться через один такт после начала выполнения предыдущей операции. Применение конвейерной организации вычислительного процесса теоретически позволяет увеличить скорость обработки данных, определяемых количеством ступеней (глубина) конвейера [1]. Реально такое ускорение вычислительной системы достигнуто быть не может по ряду причин, основной из которых является невозможность выделения в цепочке вычислений операций (подфункций конвейеризируемой функции) с примерно равными длительностями их выполнения. Принцип конвейеризации. Конвейеризация — это такой метод организации вычислительного процесса, в результате применения которого в вычислительной системе обеспечивается совмещение разных действий по выполнению базовых функций путем их разбиения на подфункции [2]. При этом за основу берутся следующие принципы:

- выполнение функции эквивалентно некоторой последовательности выполнения подфункций;
- данные, являющиеся входными для любой подфункции, являются выходными данными для предыдущей подфункции;
- никаких других взаимосвязей, кроме обмена входными и выходными данными, между подфункциями нет;
- интервалы времени, необходимые для реализации этими аппаратными блоками своих подфункций, примерно равны.

Аппаратные средства, необходимые для выполнения любой из этих подфункций, называются ступенью конвейера. На вход ступени конвейера данные поступают в дискретные моменты времени. Интервал времени между двумя соседними дискретами загруженного конвейера называется циклом конвейера. Можно выделить два уровня конвейеризации — на базе конвейерных функциональных устройств и на основе конвейерных вычислителей (КВ).

Первый из выделенных уровней предусматривает выполнение некоторой функции как последовательности подфункций и ориентирован на возможность реализации этой функции одним ФУ, организованным по конвейерному принципу. Примеры конвейеризации на первом уровне приведены в [1].

Второй уровень конвейеризации является более общим и ориентирован на конвейеризацию последовательности операций, которые могут быть самыми разными как по своей сущности, так и по времени реализации, причем каждая из операций может выполняться собствен-

ным функциональным устройством (ФУ). Такой подход к конвейеризации вычислительного процесса был предложен автором данного исследования и наиболее подробно рассмотрен в [3].

Основные особенности этого подхода заключаются в следующем:

- для уменьшения разницы во времени выполнения операций разными ступенями на реализацию одной ступенью ориентируется последовательность операций (а не одна подфункция), т. е. каждый из уровней конвейера интегрирует ряд последовательно выполняемых операций алгоритма;
- каждая из ступеней конвейера строится как последовательное соединение ФУ различного назначения;
- входные данные ступени конвейера можно получать как от предыдущей ступени, так и от других технических средств системы, в том числе и от других конвейеров.

Назовем конвейер, реализующий данные принципы, в отличие от традиционного конвейерного вычислителя (КВ) [4], интегрированным конвейерным вычислителем (ИКВ).

Конвейеризация второго уровня представляет собой гибкий инструмент при выполнении сложных алгоритмов и построении соответствующих вычислительных структур, является одним из эффективных путей выполнения требований реального времени.

При этом существенно повышается выигрыш в производительности по сравнению с традиционным КВ (ускорение) за счет повышения загруженности ступеней конвейера. Отметим, что любой из рассмотренных уровней конвейеризации позволяет получить скорость обработки потока данных, определяемую циклом конвейера.

Предложенный принцип реализации конвейера на интегрированном конвейерном вычислителе основан на возможности объединения на одном уровне ряда последовательно выполняемых подфункций с учетом их реализации в реальном времени, что позволяет максимально загрузить ступени конвейера и приблизить реальное повышение производительности конвейера по отношению к последовательной вычислительной системе к теоретической, равной глубине конвейера.

ЛИТЕРАТУРА

1. Коуги, П. Архитектура конвейерных ЭВМ. М.: Радио и связь. — 1985. 567 с.
2. Закревский, А. Д. Логический синтез каскадных схем. М.: Наука, 1981. — 414 с.
3. Кобайло, А. С. Базовый алгоритм синтеза вычислительных структур реального времени / А. С. Кобайло, Н. А. Жилияк // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. — 2007. Вып. XV. — С. 147-150.
4. Воеводин В. В. Параллельные вычисления. Издательство БХВ-Петербург, — 2002. — 609 с.

СПОСОБЫ КОДИРОВАНИЯ СОСТОЯНИЙ ЦИФРОВЫХ АВТОМАТОВ

Кожемякин Е. А., Дубяга С. П., Курулёв А. П.

Кафедра электроники, кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: efim.kozhemiakin@gmail.com

Данный материал представляет краткое описание проблемы в теории цифровых автоматов и объясняет один из способов решения данной проблемы, который был найден при попытке автоматизации процесса построения цифровых автоматов.

ВВЕДЕНИЕ

Автомат – система механизмов, устройств, в которой полностью автоматизированы процессы получения, преобразования, передачи энергии, материалов, информации.

Термин «автомат» в основном используется в двух аспектах:

- техническом;
- математическом.

При математическом подходе под автоматом понимается математическая модель, у которой должны быть входы, внутренние состояния и выходы. Детали структуры устройства не учитываются и не рассматриваются.

В техническом подходе под автоматом понимается вполне реальное устройство, например, телефонный автомат, торговый автомат и т. д. В данном случае, естественно, известными являются детали внутреннего строения устройства.

С точки зрения сигналов цифровой автомат (ЦА) – система, которая может принимать входные сигналы, под их воздействием переходить из одного состояния в другое, сохранять его до прихода следующего входного сигнала, выдавать выходные сигналы.

В данной работе рассматриваются цифровые сигналы и двоичная логика на базе логических элементов.

I. ПРИМЕНЕНИЕ

Теория автоматов лежит в основе всех цифровых технологий и программного обеспечения. Часть математического аппарата теории автоматов напрямую применяется при разработке лексических и синтаксических анализаторов для формальных языков, в том числе языков программирования, а также при построении компиляторов и разработке самих языков программирования, описания аппаратуры, а также разметки.

Другое важнейшее применение теории автоматов – математически строгое нахождение разрешимости и сложности задач.

Автоматы Мура и Мили широко применяются при проектировании цифровых устройств на основе программируемых логических интегральных схем (ПЛИС). Наличие минимальной выходной задержки, связанной с переключением выходного регистра, отсутствие нестабильности переходного процесса на выходе автомата, отсутствие сквозного распространения сигнала через комбинационную схему от входа до выхода автомата, простота описания на языках описания аппаратуры делает автомат Мура практически незаменимым. Также автоматы Мура и взаимодействующие автоматы Мили используются в генетическом программировании.

II. ОПИСАНИЕ ПРОБЛЕМЫ

Построение цифрового автомата – довольно трудоёмкий процесс. Можно выделить следующие этапы разработки ЦА:

- Очень часто разработка ЦА начинается с реализации графа, который отражает закладываемую логику в простом и понятном для человека виде.
- Оптимизация графа – с этой задачей человек может справиться довольно быстро.
- Определение разрядности памяти. Минимальное число триггеров можно вычислить по формуле:

$$n = \text{ceil}(\log_2(S));$$

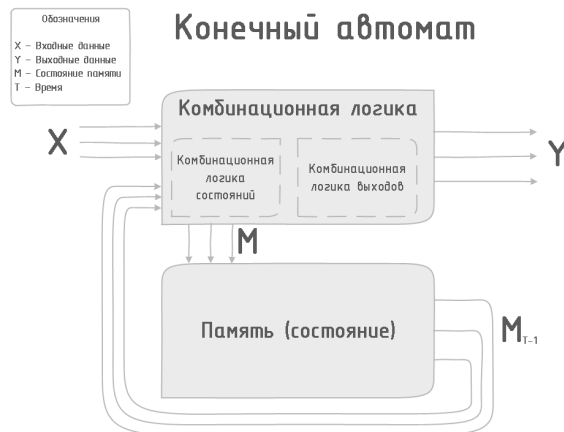


Рис. 1 – Структурно-функциональная схема цифрового конечного автомата

где, S – это число состояний, $ceil$ – функция приведения значения до ближайшего целого числа, которое не меньше исходного.

- Присвоение состояниям кодов. Алгоритма для правильного задания кодов для состояний нет. Именно от этого зависит сложность уравнений, которые мы получим для входов триггеров и количество элементов необходимых для сборки схемы.
- Составление таблицы состояний-переходов.
- Составление булевых арифметических уравнений для входов триггеров. Карты Карно составляются по таблице состояний-переходов, уравнения минимизируются.
- Преобразование уравнений для согласования с элементной базой.
- Разработка электрической схемы.

Основная проблема – отсутствие алгоритма для задания кодов состояниям автомата таким образом, чтобы уравнения для входов триггеров были как можно проще.

III. РЕШЕНИЕ

Была разработана программа для построения цифровых автоматов. На вход программа принимает граф. В программе граф представляется в наборе вершин и рёбер (вершина, входной сигнал, вершина для перехода). Итерируясь по рёбрам составляются таблицы истинности для каждого разряда в СКНФ и СДНФ. Методом Куайна-Мак-Класки минимизируются обе формы уравнений. Для каждого разряда выбирается выражение с минимальным количеством логических операций «И», «ИЛИ». Общее количество этих операций является критерием качества данной кодировки.

Количество возможных вариантов задания состояний можно рассчитать зная разрядность памяти (M) и количество состояний (S).

Количество кодов:

$$C = 2^M;$$

Количество вариантов выборки (V) нужного количества состояний (S) из всего количества кодов (C), формула из комбинаторики:

$$V = \frac{C!}{(C - S)! \cdot S!};$$

Количество возможных вариантов задания состояний (A) равно:

$$A = S! \cdot V = \frac{C!}{(C - S)!};$$

Если перебирать все варианты и потом отобрать лучший, то в зависимости от графа программа может выполняться слишком долго. Такой вариант подойдёт для автоматов с небольшим числом вариантов задания кодов состояний.

Для автоматов с большим числом возможных вариантов задания состояний был разработан генетический алгоритм перебора вариантов состояний.



Рис. 2 – Схема генетического алгоритма

IV. РЕЗУЛЬТАТЫ

Для исследования был спроектирован автомат с числом возможных вариантов задания состояний равным 6720. Для каждого варианта было рассчитано количество необходимых элементов для реализации.

Для любой выборки (V) нашлось не менее X ($X < S!$) перестановок с наилучшим исходом. Наилучший исход – исход с минимальным числом элементов необходимых для реализации данного автомата. Для поиска способа кодирования с наилучшим исходом достаточно перебрать $S!$ вариантов.

Анализ показал, что наибольшая вероятность встретить автомат с наилучшим исходом – если количество 0 и 1 в кодах состояний будет равнозначным.

Для сложных автоматов, где перебор занимает много времени, эффективным решением будет применить генетический алгоритм, он не обязательно найдёт наилучший исход, но позволит быстро найти решение близкое к нему.

СПИСОК ЛИТЕРАТУРЫ

1. Теория автоматов: Учебное пособие. Триханов А. В. – Томск: Изд. ТПУ, 1999. – 103 с.

АЛГОРИТМЫ РАСПОЗНАВАНИЯ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ В ЗАДАЧАХ КОМПЬЮТЕРНОЙ АВТОМАТИЗИРОВАННОЙ ДИАГНОСТИКИ

Козарь Р. В.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: pozitr0n.kozarroman@gmail.com

В работе представлены результаты анализа существующих методов распознавания медицинских изображений. Предложена модификация алгоритма Собеля, учитывающая особенности медицинских изображений.

ВВЕДЕНИЕ

Медицинские изображения имеют ряд особенностей, которые следует учитывать при создании систем распознавания. Основными проблемами при работе с изображениями, сделанными с помощью оптической эндоскопии являются низкое качество и плохой контраст, наличие артефактов и засветок, обусловленных ручным способом их получения.

Проблему повышения качества требуется решать на всех этапах распознавания изображений: сегментации, выделении и анализе объектов, параметрическом описании, классификации. Классификация позволяет дать ответ на вопрос: являются ли выделенные объекты отражением патологического процесса (см. [1, с. 1]).

Установлено, что для получения качественных результатов необходимо решить следующие проблемы: улучшение качества изображения, улучшение качества распознавания отдельных элементов, выбор оптимального метода машинного обучения для последующего анализа и обработки.

I. КРИТЕРИИ ОБРАБОТКИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Определение оптимальных методов параметрического описания объектов интереса, непосредственно связаны с качеством их классификации, что определяет точность последующего анализа таких изображений с минимальной вероятностью ошибки определения патологических процессов.

Для анализа методов классификации сформулированы критерии для сравнения и обработки изображений: форма представления результата (матрица интенсивностей или набор признаков и атрибутов), анализируемый параметр, который определяет, какой из параметров изображения или его части (например, это может быть яркостная, цветовая, градиентная или текстурная информация) будет использоваться при обработке, алгоритм обработки, формат изображения.

Определяет цветность изображения, необходимую для работы алгоритма. Среди известных методов, которые можно использовать для данной задачи, имеются следующие: алгоритм Канни, алгоритм Собеля, размерность Реньи, метод локальной бинарной структуры и метод Харалика.

Также необходимо отметить, что алгоритмы Собеля и Канни получают на выходе изображение. Остальные методы – числовые данные. Параметры, используемые для анализа изображений у всех методов различные, поэтому используются различные методы обработки. Как правило, требуется предварительная обработка изображения.

II. ИСПОЛЬЗОВАНИЕ ОРИГИНАЛЬНОГО И МОДИФИЦИРОВАННОГО АЛГОРИТМА СОБЕЛЯ

В работе рассматриваются медицинские изображения, полученные с эндоскопической камеры. Обычно такие изображения имеют невысокое разрешение и малый контраст. Основная задача распознавания медицинских изображений заключается в получении четкого контура исследуемой области. Качество работы нейронной сети сильно зависит от предъявляемого ей в процессе обучения набора учебных данных (см. [2, с. 27]).

Один из наиболее точных алгоритмов выделения границ является оператор Собеля – дискретный дифференциальный оператор. Оператор позволяет вычислять приближенное значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на границе, а также ориентация границы.

Точки изображения, в которых яркость изменяется особенно сильно, называют краями. (см. [3, с. 247]). Для точного выделения краев необходимо правильное определение порога (ограничения). Для выделения контуров объ-

ектов на изображении может быть использован метод линейной фильтрации, основанный на вычислении аперидической свертки фрагмента изображения со специальным ядром в пространственной области.

Используемое для вычислений ядро определяет тип линейного фильтра. Для выделения контуров методом линейно фильтрации часто фильтр Собеля, имеющий следующий вид (1):

$$y_{ij} = ((a_{i,j} - a_{i,j+2}) + 2 \times (a_{i+1,j} - a_{i+1,j+2}) + (a_{i+2,j} - a_{i+2,j+2}))^2 + ((a_{i,j} - a_{i+2,j}) + 2 \times (a_{i,j+1} - a_{i+2,j+1}) + (a_{i,j+2} - a_{i+2,j+2}))^2)^{\frac{1}{2}} \quad (1)$$

Градиентные фильтры могут быть также записаны в матричной форме. Для окна размером 3x3 маска фильтра Собеля для осей ox и oy имеют следующий вид (2):

$$ox = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} oy = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2)$$

Существует такой подход к решению задач распознавания как усиление простых классификаторов (см. [4, с. 21]). Фильтр Собеля основан на обработке изображения специальными масками для различных осей, действие которых фактически равносильно дискретному дифференцированию функции распределения интенсивностей пикселей, находящихся под ней. Матрицы для осей взаимно транспонированные. Изображение представляет собой матрицу интенсивностей пикселей. Первая матрица собирает информацию вокруг пикселя a на строке i и строке j изображения в виде следующих сумм (3),(4):

$$X = \sum_{k=1}^3 \sum_{m=1}^3 A_{1,km} b_{km} \quad (3)$$

$$Y = \sum_{k=1}^3 \sum_{m=1}^3 A_{2,km} b_{km} \quad (4)$$

Аналогичную сумму, имеющую смысл дискретной производной по направлению, ортогональному первому собирает вторая транспонированная матрица. Для более качественного распознавания медицинских изображений предлагается использовать модифицированный диагональный вариант масок. Для осей ox и oy альтернативные маски будут иметь следующий вид (5):

$$ox = \begin{bmatrix} -1 & -2 & 0 \\ -2 & 0 & 2 \\ 0 & 2 & 1 \end{bmatrix} oy = \begin{bmatrix} 0 & 2 & 1 \\ -2 & 0 & 2 \\ -1 & -2 & 0 \end{bmatrix} \quad (5)$$

Выделение дискретных производных здесь производится не вдоль осей ox и oy , а под углом

45 градусов к ним. Матрицы при этом не будут являться взаимно транспонированными. Следовательно, интенсивность линий в модифицированном фильтре больше, а точность контура выше. Чем выше точность контура – тем более достоверной будет выделенная граница. При расчете, для более точного сопоставления результатов, можно внести корректировочный коэффициент $4/5$ для всех пикселей изображения. За меру сравнения качества фильтра Собеля с различными видами масок можно принять параметр концентрации, имеющий смысл среднего минимума интенсивности по h строкам. Также отметим, что число столбцов матрицы интенсивности изображения равно w (предполагаем, что черный цвет имеет интенсивность равную нулю) (6):

$$\gamma = \frac{1}{h} \sum_i^k \max_{j=1...w} a_{ij} \quad (6)$$

Опыт показывает, что для медицинских изображений фильтр с матрицей маски диагонального вида имеет лучшие (меньшие) показатели по сравнению с оригинальной матрицей-фильтром. Для медицинских изображений, шириной (пикселей) $w = 100$, $\gamma_{(orig)} = 0.165$, $\gamma_{(mod)} = 0.088$. Для медицинских изображений большого размера, шириной $w = 400$, $\gamma_{(orig)} = 0.082$, $\gamma_{(mod)} = 0.0215$.

ЗАКЛЮЧЕНИЕ

Предложена модификация алгоритма Собеля для медицинских изображений. Высокий уровень избыточности продифференцированного медицинского изображения доказывает появление избыточного количества границ. Значения фильтра матрицы маски указывают на возможность более четкого определения границ исследуемой области. Данным условиям удовлетворяет модифицированный алгоритм Собеля, что доказывает его эффективность при выделении контуров для медицинских изображений.

СПИСОК ЛИТЕРАТУРЫ

1. Нейросетевая классификация медицинских изображений на основе спектра размерностей Ренья : сб. науч. ст. / Новосиб. гос. техн. ун-т ; под ред. А. Н. Бондаренко, А. В. Кацук. – Новосибирск : НГТУ, 2005. – 6 с.
2. Каллан, Р. Основные концепции нейронных сетей / Р. Каллан. – М.: Изд. дом «Вильямс», 2003. – 287 с.
3. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М.: Изд. Дом «Вильямс», 2004. – 928 с.
4. Технические науки: проблемы и перспективы : материалы IV Междунар. науч. конф., Санкт-Петербург, июль 2016 г. / А. А. Цветков [и др.] ; под ред. А. А. Цветкова. – СПб.: «Свое издательство», 2016. – 134 с.

УПРАВЛЕНИЕ ДИНАМИКОЙ СЛОЖНОЙ СИСТЕМЫ

Кузнецов М. В., Герман О. В.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: {kuznetsov, german}mishaky@mail.ru,

Рассматривается общий подход к построению управления динамикой сложной системы. Особенность подхода состоит в том, что он строится практически исключительно на использовании вложенной структуры кластеров, так что управление реализуется за счет погружения от одного уровня кластеров к другому с принятием решений в пределах каждого уровня кластеров.

Введение

Предлагается концепция иерархии кластеров для реализации управления сложной системой, например, финансовой, производственной, транспортной и т.п. Для целей описываемого подхода существенное значение имеет таблица наблюдений за определенный период времени о функционировании системы. На основании этой таблицы строится система вложенных кластеров, с помощью которых и реализуемой техники выбора кластеров решается задача выбора управляющего воздействия для наблюдаемого состояния системы. При этом существенными механизмами являются выбор кластера на основании используемой метрики сходства и оценка близости на основании функции минимального штрафа байесовской классификации. Представленный материал составляет некую общую методологическую основу для реализации управления сложной системой на основании концепции, известной как управление на основе прецедентов.

I. Описание подхода

Отправной позицией нашего подхода является таблица наблюдений за динамикой сложной системы (например, финансовой, производственной, транспортной и т.п.) Система описывается вектором параметров $V(t)$, состояния векторов изменяются в результате реализуемых управлений $U(V, t)$, так что поведение системы описывается достаточно сложной траекторией. Мы принимаем (хотя это не является ограничением), что всегда принимается одно какое-нибудь управление. Например, при управлении рисками примерами управляющих воздействий могут быть продажа активов, инвестирование в производство и технологии, сокращение персонала, техническое перевооружение, сокращение объемов производства и др. [1]. Для учета динамики нам можно воспользоваться наблюдениями по разным фирмам за достаточно длительный период времени. Такие данные можно описывать многомерными массивами $V[t, n, Z_t]$, где $t = 0, 1, 2, \dots, T$ задает моменты наблюдения (дискретные величины),

Z_t определяет значения наблюдаемых критериев [2].

Имея таблицу наблюдений, мы можем выделить кластера – совокупности близких значений векторов $V(t)$ [3, 4]. Число кластеров задаем равным 4. Эти кластера будут иметь такой смысл: (1) нормальное функционирование (штатный) режим; (2) режим близкий к нормальному; (3) аварийный режим (катастрофа) и (4) опасный режим – дрейф в сторону катастрофы. Разумеется, таблица с исходными данными должна содержать достаточное число представителей всех этих режимов. В минимальном варианте число кластеров можно ограничить тремя, исключив кластер, соответствующий аварийному режиму, поскольку этот режим на практике возникает в редких случаях (например, предприятие становится банкротом в единичных случаях). Разбиение на заданное число кластеров – это хорошо разработанная задача анализа данных (data mining). Можно отметить например метод k -средних для ее решения. Решается следующая задача. Наблюдается текущее значение вектора параметров $V_i(t)$. Нужно определить, какое управление реализовать для этого значения $V_i(t)$. Эта задача решается в несколько этапов. На первом этапе определяется, к какому кластеру отнести $V_i(t)$. Разумеется, можно исходить из того, что каждый кластер описывается как множество реализаций многомерной случайной величины. Тогда можно говорить о задаче отыскания вероятности принадлежности к кластеру. Однако закон распределения многомерной случайной величины а priori не известен. Кроме того, параметры могут быть коррелированы, накопленных статистических данных недостаточно, а значения параметров могут быть зашумлены. Очевидно, следует найти другую рациональную оценку. Будем использовать функцию штрафа за неправильную классификацию [5] в виде

$$F_i = \sum C_{ik} \cdot P_k \cdot P(V|k).$$

Здесь F_i – это штраф за отнесение объекта V к кластеру i ; P_k – вероятность выбора кластера k ; $P(V|k)$ – условная вероятность появления объекта V в кластере k . На практике полагают, что $C_{ik} = 1$, если $i \neq k$, и $C_{ik} = 0$ в против-

ном случае. Вероятности P_k заменяют частотами появления кластеров. Поскольку мы определяем кластера, то их частоты пропорциональны числу членов в этих кластерах (т.е. мощности кластеров). Иначе обстоит дело с $P(V|k)$. Вместо $P(V|k)$ будем использовать величины, заменяющие их. А именно: полагаем, что вероятность принадлежности к кластеру обратно пропорциональна расстоянию до него. Расстояние от объекта до кластера можно рассчитать, например, на основе формулы Евклида или Махалонобиса [6], или какой-нибудь иной известной формулы. Таким образом мы в состоянии оценить штрафы за неправильную классификацию. Далее мы относим наблюдаемый вектор $V_i(t)$ в тот кластер, для которого вычисленная оценка штрафа минимальна. Таким образом первая задача решена. Вторая задача заключается в выборе управления в границах определенного кластера. Разумеется, в рамках одного и того же кластера могли быть реализованы различные управления и даже одно и то же по сути управление могло быть реализовано различным способом. Поэтому опять сталкиваемся с задачей принятия решения [7]. Сначала следует определить, какое управление реализовать. Теперь мы должны в рамках нашего кластера задать подкластеры векторов $V_i(t)$ по различным управлениям. Теперь это делается автоматически, поскольку каждый “подкластер” однозначно определяется тем управлением, которое в нем реализовано. Наконец, мы выходим на последнюю задачу: как реализовать выбранное управление. Здесь речь идет о фазификации нечеткой лингвистической переменной управления U_i [8]. Например, можно говорить о необходимости дополнительных инвестиций. Однако объем инвестиций не определен. Кроме того, наличные резервы ограничивают размер инвестиций. Поэтому можно вести речь о незначительных инвестициях, средних инвестициях и крупных инвестициях. Нам нужно фазифицировать лингвистические значения, например, полагая, что незначительные инвестиции определены в диапазоне нечетких значений [0.1 – 0.4], средние по размеру инвестиции определены в диапазоне (0.4 – 0.66] и значительные – в диапазоне (0.66 – 1.0). Исходя из этого, опять получаем разбиение на подкластеры и снова решаем задачу классификации. Очевидно, мы сохраняем ту же идею, что и выше – а именно: использование байесовской оценки штрафа за неправильную классификацию. Строго говоря, нужно уточнить значение нечеткой переменной управления после выбора подходящего кластера либо использовать сред-

нее интервальное значение. Например, мы выбрали подкластер [0.1 – 0.4]. Среднее интервальное значение в нем равно 0.25. Наконец, остается выполнить дефазификацию найденного нечеткого значения. Это в свою очередь зависит от определения функции принадлежности.

II. ВЫВОДЫ

Особенность описанного подхода состоит в возможности его полной автоматизации, за исключением сбора и подготовки таблиц с исходными данными. Составление таких таблиц следует выполнять за достаточно продолжительный период, чтобы обеспечить объективность формируемых решений. Разумеется, интересным является вопрос о работе с недостоверными данными наблюдений и даже необходимости восстановления отсутствующих данных. Этот вопрос является предметом обсуждения [9]. Для целей автоматизации описанного подхода весьма перспективным выглядит использование языка Python. Настоящая работа лишь указывает направление исследований, оставляя в качестве выбора как конкретные методологии, так и программные и математические средства. Поскольку исходные данные играют весьма существенную роль для качественного процесса управления системой, то вопросы подготовки данных (хотя и остались за пределами данной статьи) играют весьма существенную роль в реализации изложенного подхода.

1. Hopkin, P. Fundamentals of risk management: understanding, evaluating and implementing effective risk management / P. Hopkin// –Report of the Institute of Risk Management, England.2010. –352p.
2. Han, J. Data Mining. Concepts and techniques / J. Han, M. Camber, J. Pei// Elsevier, 2012. –740p.
3. Abonyi, J.Cluster analysis for Data Mining and System identification /J. Abonyi, B. Feil. //Berlin:BirkHauser, 2007. –304p.
4. Hoppner, F.Fuzzy cluster analysis /F. Hoppner, F. Klawonn, R. Kruse/John Willey Sons, 1999. –290p.
5. Theodoridis, S. Pattern recognition/S. Theodoridis, K. Koutroumbas//Elsevier, 2003. –700p.
6. Patrick, J.The Past, Present, and Future of Multidimensional Scaling/ J. Patrick, F. Groenen, I. Borg//USA:Econometric Institute Report EI, 2013. –25p.
7. Alter, S. Decision Support Systems: Current Practice and Continuing Challenges/Addison-Wesley Pub., 1980. –316p.
8. Mendel, J. Fuzzy logic systems for engineering: a tutorial/J. Mendel//Proceedings of the IEEE, vol.83(3), 1985. –p.p.345–377.
9. George, H.Robust Decision Trees: Removing Outliers from Databases/H. George// Proceedings of the AAAI Conference on artificial Intelligence. USA, 2018. –p.p. 175-181.

АЛГОРИТМЫ И МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОЦЕНКИ УСТОЙЧИВОСТИ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ К КРИПТОГРАФИЧЕСКИМ АТАКАМ

Кузьма Ю. В., Хлопцев А. А.

Факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: dzotcrew@gmail.com, redleonfire@yandex.by

В статье рассмотрена возможность применения методов и алгоритмов машинного обучения для оценки устойчивости физически неклоняемых функций (ФНФ) к криптографическим атакам. Интерес к данной тематике обусловлен повышением риска криптографических атак, связанным с широким распространением устройств Интернета вещей и различных токенов, смарт-карт, банковских карт, интегральных схем, использующих ФНФ для построения неклоняемых идентификаторов и генерирования случайных числовых последовательностей.

ВВЕДЕНИЕ

Повсеместное распространение устройств Интернета вещей и различных токенов, смарт-карт, банковских карт, интегральных схем обусловило актуальность задачи их надёжной идентификации [1-2]. Одним из актуальных и активно развивающихся способов для генерации случайных числовых последовательностей, ключей и невоспроизводимых идентификаторов является использование ФНФ.

По определению, данному в работе [3], физически неклоняемой функцией (от англ. Physical Unclonable Function, PUF) является характеристика физической (цифровой) системы, которая не поддается клонированию (копированию, воспроизведению) на других системах. Данное свойство цифровой системы обусловлено недостаточной точностью производства и/или намеренным использованием материалов с неоднородной структурой (например, использование пасты с частицами феррита бария, различающихся по форме и размеру, при производстве магнитного носителя карт с магнитной полоской).

Широкое применение физически неклоняемых функций повышает заинтересованность злоумышленников в осуществление успешных криптографических атак на них, с целью воспроизведения (подделки), подмены исходной ФНФ. Целью данной работы является исследование возможности применения алгоритмов и методов машинного обучения для оценки устойчивости ФНФ к криптографическим атакам.

I. ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫЕ ФУНКЦИИ

Формально ФНФ описывается значениями пар входных и выходных векторов, которые в цифровых системах представляют входные C (Challenge) и выходные R (Response) сигналы. ФНФ может быть описана множеством всевоз-

можных пар запрос-ответ (Challenge-Response Pairs, CRP), а так же функцией преобразования множества i , во множество R_i :

$$R_i = PUF(C_i) \quad (1)$$

Существует множество видов ФНФ, для использования в целях идентификации важным свойством ФНФ является стабильность ответа ФНФ на многократно повторяющийся запрос при одних и тех же условиях. По этим параметрам для исследования выбрана ФНФ типа «арбитр» (АФНФ) [4]

II. ВЫБОР ИНСТРУМЕНТОВ

Авторами, данной работы, для проведения исследования были выбраны следующие инструменты:

- язык программирования Python;
- математическая библиотека numpy;
- библиотека для обработки и анализа данных pandas;
- библиотека машинного обучения scikit-learn;
- библиотека визуализации данных matplotlib.

Поскольку библиотека scikit-learn предоставляет широкий выбор моделей и различных нейронных сетей, авторами были выбраны некоторые из них, а именно модели Perceptron и SGDRegressor, а также нейронная сеть MLPClassifier

III. РЕЗУЛЬТАТЫ

Выбранные модели и нейронные сети были обучены на выборках различных размеров (1000, 5000, 10000, 50000, 100000) содержащих векторы признаков различной длины (8, 16, 32, 64, 128 бит) и значения эталонных классов для них. Ниже представлены результаты оценки точности предсказания, обученных на выборках моделей и нейронной сети, проведённой методом кросс-валидации для

Perceptron (см. рис. 1), SGDRegressor (см. рис. 1), MLPClassifier (см. рис. 1).

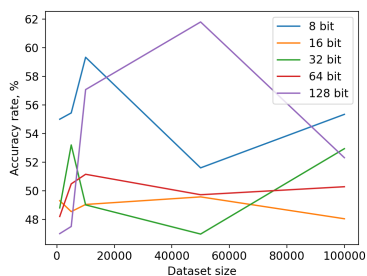


Рис. 1 – Точность предсказания Персептрон

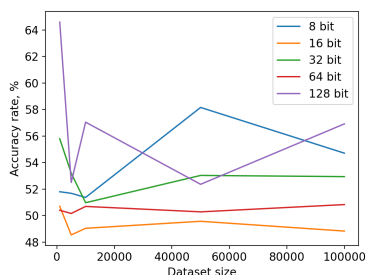


Рис. 2 – Точность предсказания SGDRegressor

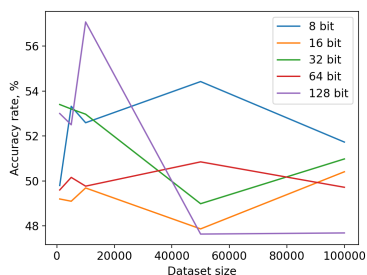


Рис. 3 – Точность предсказания MLPClassifier

Как видно из графиков, на исходных данных линейные модели и нейронная сеть не показывают значительных результатов, поэтому, основываясь на данных работы [5], было принято решение преобразовать входные данные с помощью функции Z и повторить эксперимент.

$$Z(\omega_i) = \begin{cases} \omega_i = 1 & i = 1 \\ \omega_i = \omega_{i-1} * -1^{\omega_i} & i > 1 \end{cases}$$

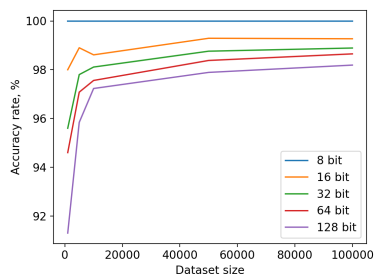


Рис. 4 – Точность предсказания Персептрон

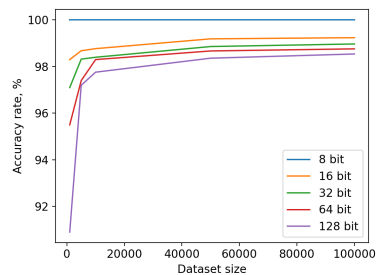


Рис. 5 – Точность предсказания SGDRegressor

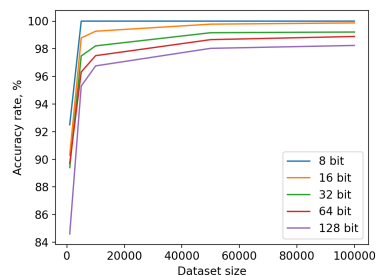


Рис. 6 – Точность предсказания MLPClassifier

Как видно из графиков (см. рис. 4-6), на преобработанных данных линейные модели и нейронная сеть показывают положительные результаты.

IV. ЗАКЛЮЧЕНИЕ

В ходе исследования были получены положительные результаты применения алгоритмов и методов машинного обучения для оценки устойчивости АФНФ к атакам, осуществляемым с помощью машинного обучения и необходимости разработки метода защиты функции от атак такого типа.

1. London Calling: Security technology takes time. UBM Tech Electronics [Electronic resource] / Peter Clark – EE Times, 2013. – Mode of access: <https://www.eetimes.com/nxp-and-intrinsic-id-to-raise-smart-chip-security/>. – Date of access: 12.10.2020
2. NXP and Intrinsic-ID to raise smart chip security [Electronic resource] / EETimes – EE Times, 2010. – Mode of access: <https://www.eetimes.com/nxp-and-intrinsic-id-to-raise-smart-chip-security/>. – Date of access: 12.10.2020
3. Architecture and Design Flow for a Highly Efficient Structured ASIC / H. Man-Ho // IEEE Transactions on VLSI Systems. –2012. – Vol. 21, iss. 3. – P. 423-433
4. A technique to build a secret key in integrated circuits for identification and authentication applications / J. W. Lee // Intern. Symp. VLSI Circuits (VLSI'04), Honolulu, USA, June 15–19, 2004 –Honolulu, 2004. – P. 176–179.
5. Клыбик, В. П. Метод увеличения стабильности физически неклонированной функции типа «арбитр» // В. П. Клыбик, С. С. Заливако, А. А. Иванюк // Информатика –2017. – № 1. – С. 32–36.

ПРИМЕНЕНИЕ ВЕГЕТАЦИОННЫХ ИНДЕКСОВ В СИСТЕМЕ МОНИТОРИНГА ПОЖАРООПАСНОЙ ОБСТАНОВКИ В ЛЕСАХ

Лабоха А. К., Шамына А. Ю., Ардяко А. Д.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: labokha.poit@bsuir.by, shamyna@bsuir.by, ardyakon@gmail.com

Проанализированы недостатки существующих методик определения уровня пожароопасности в лесах, изучен зарубежный опыт. Рассмотрены возможности использования вегетационных индексов при определении пожарной опасности в лесах на примере Воложинского лесхоза.

ВВЕДЕНИЕ

В настоящее время во многих сферах жизни используется принцип предупреждения и минимизации последствий потенциально опасных ситуаций. Это обусловлено тем, что в большинстве случаев возможный ущерб может многократно превысить затраты на оперативный мониторинг ситуации, построение прогнозов и профилактику опасных ситуаций.

Такая ситуация справедлива и для лесного хозяйства: его проблемой являются пожары, возникающих на лесных территориях. Отсутствие системы оперативного мониторинга пожароопасной обстановки может привести к неконтролируемому распространению лесного пожара (особенно при неблагоприятной метеорологической обстановке), уничтожению лесной экосистемы, а также угрозе жизни и здоровью человека в случае пожара вблизи населенных пунктов. Системы мониторинга позволяют более эффективно и локально проводить мероприятия по предотвращению распространения пожара, снижать факторы, а построение прогноза с учетом прогноза погоды позволит более комплексно оценивать пожароопасную обстановку.

I. МЕТОДИКИ ОПРЕДЕЛЕНИЯ ПОЖАРООПАСНОСТИ

Используемая в настоящее время в Беларуси методика определения пожарной опасности в лесах (СТБ 1408-2003) использует комплексный показатель Диченкова, который учитывает исключительно фактическую метеорологическую обстановку. Вопрос увеличения пространственного разрешения метеорологических данных и построения прогноза может быть решен путём использования данных прогнозных моделей погоды (GFS, WRF и др.). Однако фактически на вероятность возникновения пожара также оказывает структура лесного фонда и его текущее состояние на момент изучения обстановки.

С развитием спутниковых систем появилась возможность оперативной оценки указанных факторов на основе аэрокосмических снимков благодаря их доступности, актуальности и

высокому разрешению. Выполненная канадскими исследователями работа по оценке пожарной обстановки с использованием данных ДЗЗ [1] продемонстрировала обоснованность данного направления исследований; для определения уровня пожароопасности использовались полученные на основе данных ДЗЗ нормализованные вегетационные индексы NDVI и NDWI, температура поверхности Земли, а также суммарное накопленное количество влаги в результате осадков и влажности.

Оценка состояния и динамики изменений лесного фонда осуществляется с помощью вегетационных индексов, которые рассчитываются как различные комбинации спектральных каналов аэрокосмических снимков высокого разрешения. Вегетационные индексы позволяют оценить параметры растительности, причем именно отклонения значений в пределах одной территории позволяют выявлять участки в особом состоянии. Однако ввиду новизны сферы исследования не существует единого выработанного набора вегетационных индексов, который бы гарантировал качественную оценку состояния растительности. Так, ученые из Ирана при проведении схожего исследования [2] использовали в значительной степени отличающийся набор вегетационных индексов. Поэтому, ввиду отличия климатических условий в Беларуси, Канаде и Иране, одной из задач исследования является экспериментальный подбор оптимальной комбинации вегетационных индексов для определения пожароопасной ситуации для территории Беларуси.

II. ИСПОЛЬЗОВАНИЕ АЭРОКОСМИЧЕСКИХ СНИМКОВ

В качестве источников аэрокосмических снимков были выбраны спутниковые системы Sentinel-2 и Landsat-8. Одной из основных проблем при использовании снимков космических аппаратов является наличие облачности на снимках; и если наличие небольшой «дымки» и искажение цветности по этой причине является исправимым, то в случае сплошной или преимущественной облачности снимок может да-

вать малое количество полезной информации, что нельзя исправить с помощью методов постобработки. В условиях, когда один и тот же спутник может выполнять съемку местности с периодичностью порядка 8 суток, наличие сплошной облачности при выполнении двух повторных снимков будет приводить к временной разнице между снимками больше 20 суток. Для систем оперативного реагирования подобный срок является слишком большим, поскольку использование устаревших данных может привести к принятию ошибочного решения. По этой причине проблемы облачности и периодичности облета тесно связаны.

Использование данных нескольких компьютерных миссий может значительно повысить эффективность разрабатываемой системы, поскольку выполнение снимков в другой день как повышает вероятность получения снимка без облачности, так и увеличивает периодичность обновления снимков. Альтернативным решением проблемы облачности на снимках является разработка алгоритма, который позволил бы заменить «загрязнённые» облаками пиксели на соседние как по дате выполнения снимка, так и по рассматриваемой территории.

III. ПОСТРОЕНИЕ ВЕГЕТАЦИОННЫХ ИНДЕКСОВ

Исследования были проведены для территории Воложинского лесхоза Минской области. Были получены контуры границ лесных кварталов и выделов, сведения о местах и времени зафиксированных пожаров для последующей верификации результатов исследования. В качестве базовой даты для начала исследования было выбрано 24 апреля 2019 года, когда спутником Sentinel-2 были выполнены снимки Воложинского района при полном отсутствии облачности. Полный снимок территории лесхоза был получен в результате совмещения четырех снимков и последующей обрезки по экстенду. Последнее необходимо для минимизации временных затрат на их обработку.

Были вычислены вегетационные индексы NDWI, PSRI, EVI, TVDI, EMCI. На рисунке 1а представлена карта вегетационного индекса

PSRI, который характеризует отражения огрубевшим углеродом растительных тканей. Однако поскольку абсолютное значение этого и других индексов не позволяет оценить пожароопасную обстановку, в соответствии с методикой исследования [1] значение в каждом пикселе карты вегетационного индекса сравнивалось со средним значением индекса на исследуемом фрагменте карты. В результате все пиксели карты вегетационного индекса были разделены на две группы с границей по математическому ожиданию (рис. 1б). В случае, если значение вегетационного индекса PSRI в точке превышает среднее значение, то можно говорить о повышенном влиянии факторов, характеризующих данным индексом; подобное утверждение справедливо для всех исследуемых индексов. Соединение полученных карт влияния N индексов позволяет получить оценку, сколько факторов риска (от 0 до N) оказывают влияние в заданной точке.

Следует отметить, что в зависимости от выбора исследуемой территории разделение карты вегетационного индекса относительно среднего значения может претерпевать значительные изменения. При исследовании малой территории (лесхоз, несколько лесничеств) большее влияние имеют локальные аномалии значений индекса; в этом случае целесообразно выделение пожароопасных участков на уровне лесных кварталов и выделов, что позволит повысить эффективность мероприятий по предотвращению возникновения пожаров для работников лесничеств. При выборе большей территории (область, страна) будет выполняться сглаживание локальных экстремумов, в результате будет получена более репрезентативная оценка пожароопасной обстановки в целом; подобный подход позволит выделить области, где необходимо введение ограничений или запрета на посещение лесов.

1. Chowdhury, E. H.; Hassan, Q. K. Development of a New Daily-Scale Forest Fire Danger Forecasting System Using Remote Sensing Data. *Remote Sens.* 2015, 7, 2431-2448.
2. Abdollahi, Sara & Ostad-Ali-Askari, Kaveh & Eslamian, Saeid & Singh, Vijay. (2018). Predicting Fire Hazard Areas Using Vegetation Indexes, Case Study Forests of Golestan Province, Iran. *Journal of Geography.* 1. 10.24294/jgc.v1i2.451.

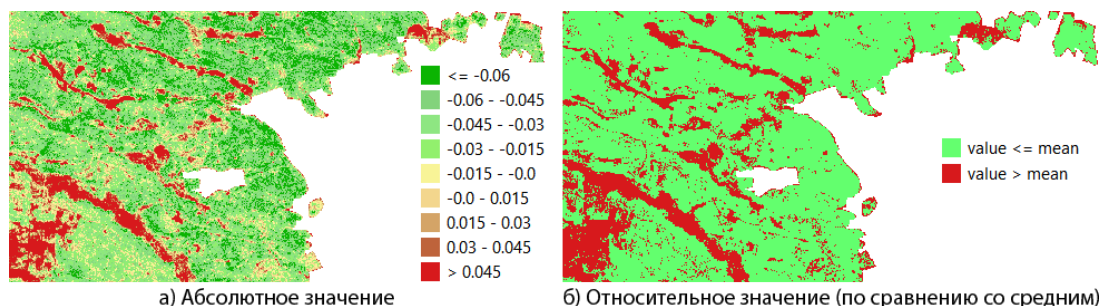


Рис. 1 – Значение индекса PSRI для фрагмента Воложинского района

ЗАДАЧА ОЦЕНИВАНИЯ КООРДИНАТ ТОЧЕК ПАДЕНИЯ АРТИЛЛЕРИЙСКИХ СНАРЯДОВ ПРИ ОБСЛУЖИВАНИИ СТРЕЛЬБЫ

Леонович А. С., Ярмолик С. Н., Свинарский М. В., Храменков А. С.

Кафедра автоматика, радиолокации и приемо-передающих устройств

Военная академия Республики Беларусь

Минск, Республика Беларусь

E-mail: {leonovich.2020}@list.ru

Рассмотрены особенности определения точек падения артиллерийских снарядов для решения задачи корректировки огня собственной артиллерии при обслуживании стрельбы. В основе определения точек падения снарядов лежит экстраполяция фильтрованных координат его траектории полета.

ВВЕДЕНИЕ

Анализ локальных вооруженных конфликтов, для которых характерна мобильная тактика ведения боя, подтверждает широкое применение противником артиллерийского огня. В условиях современного боя важно преждевременно подавить огневые позиции противника за максимально короткое время при условии обеспечения их оперативного вскрытия и точной корректировки собственного артиллерийского огня. Одним из источников информации о воздушной обстановке в зоне боевых действий являются радиолокационные средства разведки [1 С. 53]. Современные образцы радиолокаторов артиллерийской разведки способны решать задачи определения координат огневых позиций противника, мест падения снарядов, а также корректировать огонь своих огневых средств при обслуживании стрельбы. Качество функционирования средств разведки, как правило, определяется решением вышеперечисленных задач с требуемой точностью [1 С. 54].

Принцип работы радиолокатора контрбатарейной борьбы (КББ) основывается на засечке артиллерийских средств поражения на участке траектории полета и проведении нескольких измерений текущего положения боеприпаса для расчета его координат и параметров движения [1 С. 56]. Для определения места падения сопровождаемого снаряда экстраполируется нисходящий участок траектории наблюдаемого объекта (рисунок 1).

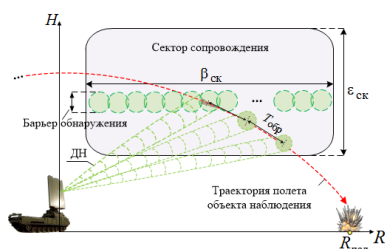


Рис. 1 – Принцип работы радиолокатора контрбатарейной борьбы при анализе нисходящего участка траектории объекта

Решение задачи экстраполяции осуществляется приближенным методом, при котором реальная траектория движения снаряда заменяется параболической кривой с приближенным учетом сопротивления воздуха.

Для засечки огневых позиций артиллерии противника луч диаграммы направленности радиолокатора сканирует пространство по азимуту (β) над линией горизонта, образуя барьер обнаружения (рисунок 1). При получении отраженного сигнала станция сопровождает цель в течение времени, ограниченного временем нахождения объекта в анализируемом секторе сопровождения ($\beta \times \epsilon$). Для организации сопровождения обнаруженного объекта луч ДНА с периодом T перестраивается в точку экстраполяции его координат. После выхода наблюдаемого объекта из анализируемого сектора радиолокатора происходит процедура вычисления координат точек падения снаряда (R) [1 С. 56].

В докладе рассмотрены особенности определения точек падения артиллерийских снарядов для решения задачи корректировки огня своей артиллерии при обслуживании стрельбы.

ОСНОВНАЯ ЧАСТЬ

Значения ошибок определения точек падения являются случайной величиной, распределенной по гауссовскому закону распределения. Одной из основных величин, характеризующих нормальный закон распределения значений ошибок определения точек падения является среднеквадратическое отклонение (СКО). Методом статистических испытаний рассчитана среднеквадратическая ошибка определения точек падения анализируемого объекта. В качестве устройства вторичной обработки использовался совместный фильтр координат и параметров движения наблюдаемого объекта, основанный на методе наименьших квадратов [2 С. 71].

Для оценки величины ошибок определения точек падения снаряда проведено математическое моделирование. В качестве примера использовались тактико-технические характеристики снаряда для гаубицы – F-3 ($D = 23$ км,

– дальность стрельбы; $d = 155$ мм, – калибр снаряда; $V_{max} = 725$ м/с, – максимальная скорость полета снаряда; $m = 43$ кг, – масса снаряда [3 С. 269]).

Результаты математического моделирования представлены на рисунке 2 (где R – начальная дальность точки старта снаряда от радиолокатора, β – начальный азимут точки старта снаряда относительно севера (оси OX), ψ – начальный курс полета снаряда относительно севера (оси OX).

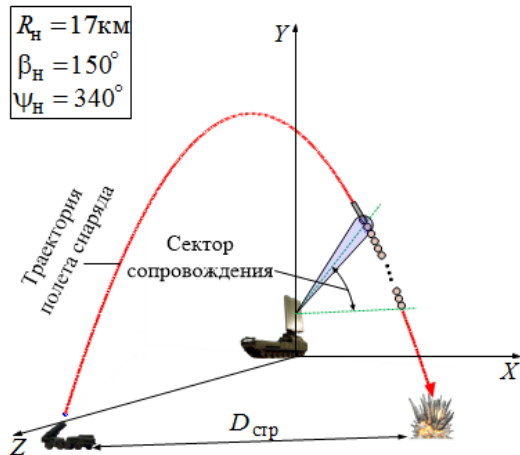


Рис. 2 – Траектория полета наблюдаемого снаряда

Точность измерения разовых оценок координат наблюдаемого объекта в процессе его полета составляла: $\sigma_r = 8.463$ м, – СКО ошибки измерения дальности; $\sigma_\beta = 0.226$ град, – СКО ошибки измерения азимута; $\sigma_\epsilon = 0.226$ град, – СКО ошибки измерения угла места. Для заданной траектории полета снаряда, радиолокатор наблюдал $N = 66$ точек (контактов) траектории. Методом статистических испытаний для оценивания СКО ошибки определения точки падения снаряда проводилось $N = 10^5$ опытов.

Результаты математического моделирования в виде оценок точек падения снаряда в горизонтальной плоскости (где X – направление на север, Z – на восток) представлены на рисунке 3.

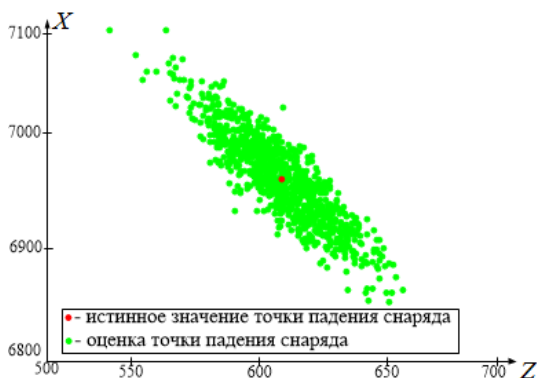


Рис. 3 – Результат определения точки падения снаряда

Результат оценки СКО ошибок определения точки падения наблюдаемого снаряда составил $\sigma = 41.4$ м. Таким образом, представленные результаты моделирования позволяют решать задачу корректировки собственного огня артиллерии при обслуживании стрельбы.

ЗАКЛЮЧЕНИЕ

В условиях постоянного развития и совершенствования средств нападения, важной и неотъемлемой частью современных систем вооружения является их способность эффективно противодействовать организованному противнику. Одним из способов противодействия огневым средствам поражения противника можно отметить корректировку собственного огня артиллерии. Представленный подход к определению точек падения снаряда позволяет решать задачу корректировки огня своих огневых средств с требуемой точностью.

1. Шишов Ю.А. Ракетно-артиллерийское вооружение / Ю.А. Шишов, А.Д. Леднев, Ю.Н. Агеев – М.: Министерство обороны СССР, 1988. –147 с.
2. Фарина, А. Цифровая обработка радиолокационной информации. Сопровождение целей / А. Фарина, Ф. Студер ; пер. с англ. А. М. Бочкарева – М.: Радио и связь, 1993. – 320 с.
3. Шунков В.Н. Артиллерия / В. Шунков – Мн.: ООО «Попурри», 2001. –704 с.

ПРИМЕНЕНИЕ ПЛАТФОРМЫ SAP HANA ДЛЯ АНАЛИЗА БОЛЬШИХ ОБЪЁМОВ ДАННЫХ

Минковский Д. В.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: dzmitry.minkouski@gmail.com

Промышленные предприятия постоянно сталкиваются с необходимостью обработки данных, которые в большом объеме поступают с заводов или других объектов. Компании собирают триллионы байт информации о своих клиентах, поставщиках и производственных операциях. Согласно исследованию консалтинговой компании IDC, к 2023 г. объем ценных данных увеличился в два раза, и 60% всех собранных данных можно будет использовать в процессе принятия решений. В статье рассматривается применение платформы SAP HANA для обработки и анализа больших объёмов данных

ВВЕДЕНИЕ

Современные системы автоматизации позволяют хранить данные с высоким уровнем детализации, что позволяет аналитикам создавать отчёты практически любой степени сложности. Однако, часто такие отчёты могут оказаться бесполезны из-за проблем с производительностью. Формирование отчета может занимать несколько часов, а иногда даже несколько десятков часов. Поэтому такие отчеты становятся неудобны, и часто от них отказываются предприятия, т.к. они не могут предоставить информацию своевременно. В таких случаях правильным решением будет использование специально разработанной для обработки больших объёмов данных платформы SAP HANA.

I. ОБЗОР SAP HANA

SAP HANA – это многоцелевое решение для хранения и обработки информации. Одна из особенностей SAP HANA – это встроенный механизм вычислений, который позволяет переносить выполнение операций по планированию с уровня приложений на уровень базы данных SAP HANA. С помощью современной архитектуры аппаратной платформы вычисления проходят эффективнее – вся «лавина» обрабатываемых данных разбивается на строго определённое количество потоков, число которых равно общему количеству ядер платформы. Такой подход позволяет максимально эффективно использовать вычислительную мощность каждого ядра каждого процессора. SAP HANA также предоставляет технологии для хранения и обработки данных in-memory. SAP HANA как база данных позволяет хранить данные в построчном и в поколонном виде. Технология хранения и обработки данных in-memory обеспечивает быструю обработку транзакций, а вместе технологией анализа данных Calculation View гарантирует высокое быстродействие при выполнении аналитических запросов [1]. Основным ядром в SAP HANA является компонент СУБД. В осно-

ве СУБД SAP HANA используется реляционная модель данных, но также существует возможность обращения к данным с помощью «графового» языка запросов WIPE. Гибкость в выборе языка запросов обусловлена архитектурными возможностями SAP HANA и заключается в использовании единого представления данных в In-Memory хранилище. Таким образом, у пользователя есть возможность обращения к данным с помощью различных семантических конструкций, используя при этом единую копию данных в памяти СУБД. Классический подход, принятый в ряде других OpenSource СУБД, отличается от вышеуказанного, потому что подразумевает использование как минимум двух хранилищ данных и разделение способа хранения графовых структур и реляционных таблиц.

II. КОНЦЕПЦИЯ УПРАВЛЕНИЯ ДАННЫМИ

На рисунке 1 отражена общая схема управления данными в SAP HANA и суть концепции управления с помощью различных языков – в частности, SQL и WIPE. Используя движок Data Processing, можно сформировать на уровне Data Manipulation новый семантический уровень для работы с данными, но при этом будет применена единая копия исходных данных, что существенно повышает возможности платформы SAP HANA для решения задач, где требуется представление информации в виде графовых структур.

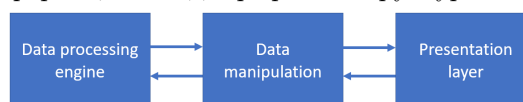


Рис. 1 – Концепция управления данными

Технология In-Memory в СУБД SAP HANA позволяет хранить и обрабатывать данные в памяти, используя уникальные алгоритмы, разработанные в компании SAP и на базе платформы Intel x86. Недавно SAP также анонсировала поддержку платформы IBM Power для SAP Hana. Уникальность и высокая скорость обработки запросов к данным заключается в возможности их

хранить и выполнять. Они находятся в сжатом виде в памяти RAM. Благодаря разработанному алгоритму обработки данных в SAP HANA удалось реализовать подход Unified Tables, который обеспечивает высокую скорость чтения и записи данных в таблицу поколочного хранения [2]. Поэтому одним из главных преимуществ SAP HANA является возможность выполнять аналитические запросы сразу на транзакционных данных, которые добавляются в реальном времени. При этом система автоматически берёт на себя обеспечение прозрачного доступа к данным. Таким образом, новые данные в таблице сразу доступны для анализа без предварительной обработки.

III. АНАЛИТИЧЕСКИЕ СЕРВИСЫ SAP HANA

В SAP HANA есть не только база данных, но и целый набор сервисов для разработки приложений, средства интеграции и очистки данных, библиотеки для аналитической обработки данных, включая Machine Learning, а также возможности для хранения и обработки специальных типов данных. SAP HANA позволяет без дополнительных инструментов загружать данные из различных источников, разрабатывать различные формы для ввода, редактирования и анализа данных. Также доступны инструменты для сложной интеллектуальной обработки данных: преобразование, трансформация, поиск закономерностей, исследования. И, конечно, платформа открыта для визуального анализа данных через различные инструменты. На рисунке 2 можно увидеть все сервисы, доступные на платформе SAP HANA [1].

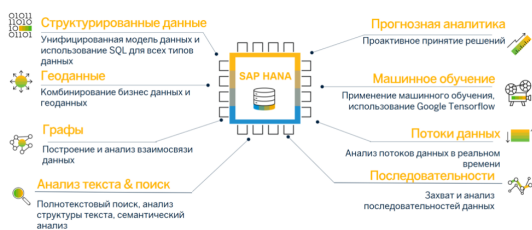


Рис. 2 – Аналитические сервисы SAP HANA

Давайте рассмотрим наиболее популярные.

SAP HANA включает в себя движок для хранения и обработки геоданных – данных, которые описывают положение, форму и ориентацию объектов в пространстве. SAP HANA поддерживает пространственные типы данных и методы их обработки. Существует специальный метод для обработки такой структуры – граф. SAP

HANA в этом случае предоставляет возможности для обработки гиперсвязанных данных и их отношений. Движок для обработки данных имеет встроенные алгоритмы поиска окрестностей, кратчайших путей, сильно связанных компонент, сопоставления образцов и многое другое.

В SAP HANA также есть сотня предварительно упакованных алгоритмов машинного обучения и прогнозирования с такими возможностями, как объединение, кластеризация, классификация, регрессия, распределение вероятности, временные ряды и многое другое [3]. Кроме этого, вы можете использовать библиотеку TensorFlow и язык R.

SAP HANA имеет встроенные возможности для обработки и анализа текстовых файлов, включая различные функции по интеллектуальному анализу текстов – например, нечеткая логика, поиск синонимов, семантический разбор и т.д.

SAP HANA Streaming Analytics может фиксировать, фильтровать, анализировать и воздействовать на миллионы событий в секунду в режиме реального времени, сохраняя данные или результаты в базу данных SAP HANA и направляя менее критические данные в более дешевые решения для хранения – такие, как Hadoop. SAP HANA Streaming Analytics также интегрирована с системой сообщений Apache Kafka[3].

ЗАКЛЮЧЕНИЕ

Крупномасштабная инфраструктура хранения, обработки и анализа данных SAP HANA делает платформу наиболее подходящим местом для загрузки, обработки и последующего хранения терабайтов менее структурированных внешних данных – потоков кликов, журналов, IoT, текста, изображений и видео. Анализ обработанных данных позволяет пользователям использовать проверенную платформу для вычисления в памяти и использовать ее высокую производительность и параллелизм для аналитики с малыми задержками.

СПИСОК ЛИТЕРАТУРЫ

1. Practical Guide to SAP HANA and Big Data Analytics / Dominique Alfermann, Stefan Hartmann // N. Engl. SAP Press. – 2018. – Vol. 44, № 8. – P. 238–243.
2. SAP HANA analytics cloud [Electronic resource] / – Mode of access: <https://help.sap.com/viewer>. - Date of access: 15.10.2020.
3. Advanced Analytics | SAP HANA 2.0 [Electronic resource] / – Mode of access: <https://blogs.sap.com/2019/11/02/advanced-analytics-sap-hana-2.0/>. – Date of access: 14.10.2020.

АЛГОРИТМЫ ОПТИМИЗАЦИИ СБОРКИ WEB-ПРИЛОЖЕНИЙ

Мазура А. А., Гуринович А. Б.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: irynamazura22@gmail.com, gurinovich@bsuir.by

Для разработчиков проектов с большими объемами программного кода является основной проблемой оптимизации сборки веб-приложений. Поэтому актуальны исследования методов сборки веб-приложений и разработки алгоритмов оптимизации данных методов.

ВВЕДЕНИЕ

JS-приложения, сайты и другие ресурсы становятся сложнее и инструменты сборки – это реальность web-разработки. Сборщики помогают упаковывать, компилировать и организовывать приложения. По мере развития и роста приложения увеличивается и время его сборки – от нескольких минут при пересборке в development-режиме до десятков минут при «холодной» production-сборке. Это оказывает влияние на продуктивность работы и в дальнейшем может значительно увеличить время разработки приложения.

Большой объем сборки замедляет загрузку приложения, что в перспективе негативно влияет на отношение пользователей. По статистике Google, если сайт загружается дольше трех секунд, 53% пользователей покинут его. Каждый второй ожидает, что страница загрузится менее чем за 2 секунды. 46% людей говорят, что ожидание загрузки страниц – это то, что им больше всего не нравится при просмотре веб-страниц [1].

I. АНАЛИЗ СБОРКИ WEB-ПРИЛОЖЕНИЯ

Первым шагом к оптимизации процесса сборки web-приложения является анализ профиля его сборки. В случае с Webpack существует инструмент [2], позволяющий визуализировать содержимое сборки и оценить размеры включенных файлов:

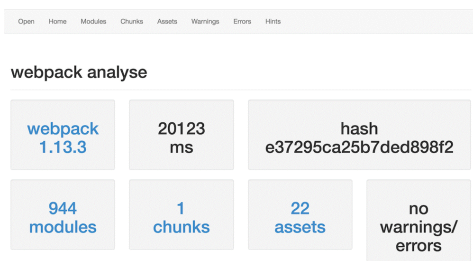


Рис. 1 – Визуализация результатов анализа профиля сборки

В данном примере на процесс сборки было затрачено около 20 секунд и скомпилировано 944 модуля в 1 файл фрагмента. Это означает, что

Webpack должен обрабатывать 944 модуля при каждой компиляции. При переходе в раздел модулей появляется график, представляющий отношения между необходимыми модулями и исходным кодом.

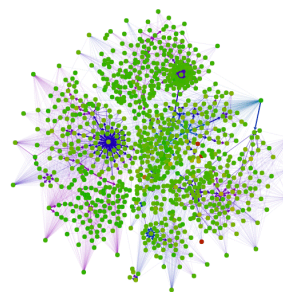


Рис. 2 – Визуализация результатов анализа профиля сборки

Webpack необходимо обработать все это дерево, чтобы собрать проект. На изображении можно идентифицировать несколько узлов с большим количеством связанных зависимостей. В большинстве случаев эти узлы являются точками входа в библиотеки, которые используются в проекте. Но зачем включать их в процесс сборки, если они не меняются? Это один из важных моментов для оптимизации сборки: нет необходимости перекомпилировать библиотеки.

II. ОПТИМИЗАЦИЯ СБОРКИ

Кеширование позволит сохранять результаты вычислений для дальнейшего переиспользования. Первая сборка может быть немного медленнее обычной из-за накладных расходов на кеширование, однако последующие будут гораздо быстрее за счет переиспользования результатов компиляции неизменившихся модулей. Причина, по которой использование исключительно кеширования не помогает, заключается в том, что кеш веб-пакетов предназначен только для повышения производительности инкрементных сборок. Его цель – ускорить обновления в режиме просмотра, а не быстро запускаться. Так что проблему оптимизации сборки полностью это не решит.

С помощью подключаемого модуля DLL webpack можно собрать node_modules только один раз и забыть о нем, пока снова что-то не

изменится в `node_modules`. В следующих сборках Webpack будет повторно использовать модули из скомпилированного `dll`-пакета. Процесс немного сложный, но оно того стоит. Сначала создается свой `dll-bundle` и `manifest.json` с отдельной конфигурацией веб-пакета и `DllPlugin`. А затем применяется `DllReferencePlugin` для повторного использования `dll-bundle` в любой другой сборке веб-пакета. Значительный плюс в том, что он работает не только с `node_modules`. Можно переместить любую часть приложения в `dll-bundle`. Единственная особенность при использовании `DllPlugin` заключается в том, что при импорте из пакета `npm`, он не знает `package.json`: `main` и сохраняет полный путь к импортированному файлу. Для разработчика это означает, что после сборки `dll-bundle` в `manifest.json` вместо путей импорта, таких как `redux`, будут найдены `./redux/es/index.js`. Это нарушит пути импорта приложения. Чтобы решить эту проблему, нужно передать измененное содержимое `manifest.json` в `DllReferencePlugin` [3].

Также добиться оптимизации процесса сборки можно с помощью минификации JS-кода, к примеру, библиотекой `UglifyJS`. `Minify` (также известный как `uglify`) в программировании - это процесс удаления всех ненужных символов из исходного кода без изменения его функциональности. Эти ненужные символы обычно включают в себя символы пробела, символы новой строки, комментарии и иногда разделители блоков, которые используются для повышения читабельности кода, но не требуются для его выполнения. Минифицированный исходный код особенно полезен для интерпретируемых языков и форматов обмена данными, развертываемых и передаваемых в Интернете (например, `JavaScript`), поскольку он уменьшает объем данных, которые необходимо передать.

```

<img alt="Screenshot of minified HTML code showing a form element with various attributes and inline styles."/>

```

Рис. 3 – Пример минифицированного кода

Минифицированный исходный код может быть отменен с помощью процесса `beautify` (так-

же известного как `prettify`), который применяется к нему различные стилистические соглашения о форматировании. Эти соглашения о форматировании обычно регулируют позиционирование, интервалы и аналогичные изменения, предназначенные для облегчения просмотра, чтения и понимания содержимого [4].

Для CSS и изображений необходимо использовать специальные плагины и загрузчики, а версионирование сборки позволит пользователю перезагружать минимально возможную часть кода.

В данной работе исследуется эффективность совместного использования алгоритмов кэширования, минификации и параллелизации процесса сборки web-приложения.

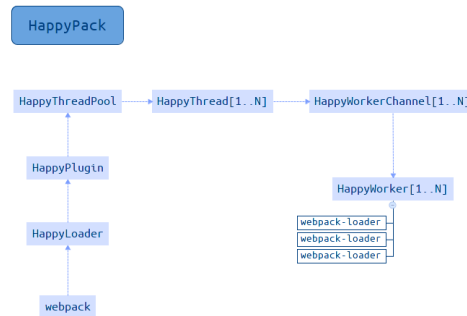


Рис. 4 – Алгоритм параллелизации потоков сборки

III. ЗАКЛЮЧЕНИЕ

Результаты анализа сборки web-приложения до и после оптимизации показали, что использование алгоритма параллелизации потоков сборки совместно с кэшированием и минификацией позволит в некоторых случаях ускорить процесс в несколько раз и значительно уменьшить размер сборки, что в дальнейшем позволит ускорить загрузку приложения на сервере.

СПИСОК ЛИТЕРАТУРЫ

1. The need for mobile speed. / Mode of access: <https://blog.google/products/admanager/the-need-for-mobile-speed/> – Date of access: 17.10.2020.
2. Webpack bundle analyzer. / Mode of access: <https://www.npmjs.com/package/webpack-bundle-analyzer> – Date of access: 17.10.2020.
3. DLL Plugin. / Mode of access: <https://github.com/webpack/docs/wiki/list-of-plugins#dllplugin> – Date of access: 17.10.2020.
4. JS Minify. / Mode of access: <https://www.uglifyjs.net/> – Date of access: 17.10.2020.

АЛГОРИТМ ИЗВЛЕЧЕНИЯ КЛЮЧЕВЫХ СЛОВ

Мазура И. А., Гуринович А. Б.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: irynamazura22@gmail.com, gurinovich@bsuir.by

Исследование алгоритмов извлечения ключевых слов включает в себя как исследование известных решений, так и поиск новых алгоритмов. Актуален анализ методов, позволяющих оптимизировать существующие алгоритмы.

ВВЕДЕНИЕ

Во многих документах и научных работах зачастую содержится большое количество текста, которое не несет существенной информации. Людям, у которых нет времени на прочтение полного текста, нужно ознакомиться с кратким содержанием.

Таким образом, появляется необходимость сокращать объем документа, выделяя наиболее значимую часть текста, называемую рефератом. Ручное реферирование является сложной и рутинной работой. Данная задача требует дополнительных сотрудников, поэтому целесообразно использовать системы автоматического извлечения ключевых слов из текста.

Наборы назначенных вручную или автоматически выделенных ключевых слов и словосочетаний из текста используются для формирования у пользователя общего представления о содержании текста. Необходимо различать два основных подхода к решению проблемы автоматизации выделения ключевых слов и словосочетаний и их извлечение. Главное отличие заключается в том, что первый подход позволяет выделять только те ключевые слова и словосочетания, которые содержатся в некотором предусмотренном словаре, а второй подход предполагает выбор ключевой информации непосредственно из текста.

В статьях ключевые слова формируют важный компонент, поскольку они обеспечивают краткое представление о содержании. Ключевые слова также играют важную роль в поиске статьи из информационно-поисковых систем и для поисковой оптимизации. Традиционные подходы к извлечению ключевых слов предполагают ручное присвоение ключевых слов на основе содержания статьи и суждений авторов. Это требует много времени и усилий, а также может быть неточным с точки зрения выбора соответствующих ключевых слов.

Так как человеческая речь иногда достаточно непредсказуема и не укладывается в четкие границы заранее прописанных правил, то данная задача должна иметь нетривиальное решение.

В работе представлена общая схема решения подобных задач. Конкретные реализации

данной схемы могут быть абсолютно разными, однако условные основные шаги будут одинаковыми.

I. КЛЮЧЕВЫЕ СЛОВА И ИХ ФУНКЦИИ

В общенаучном плане ключевые слова рассматриваются как наиболее важный, существенный для понимания смысловый элемент. Несмотря на разнообразие многосторонних подходов, которые уже были выведены, к толкованию ключевых слов, абсолютно объективным представляется то, что ключевое слово определяет содержание всего текста и является носителем его основного смысла. Учитывая это, следует признать, что ключевые слова – это слова, наиболее значимые и существенные для понимания всего содержания текста. Все основные признаки ключевых слов приведены на рисунке 1.



Рис. 1 – Основные признаки ключевых слов

Функции ключевых слов:

- ключевые слова формируют смысл текста и обеспечивают хранение его в памяти;
- ключевые слова служат созданию структурно-семантического единства текста, его целостности;
- ключевые слова задают функциональный стиль развертываемого текста (при развертывании сжатого текста с опорой на ключевые слова сохраняется стиль исходного текста);
- анализ ключевых слов позволяет извлечь целостность содержания, которое является относительно неизменным и независимым от интерпретатора;
- логически упорядоченные ключевые слова передают обобщенное содержание текста, указывают и ограничивают направление ассоциаций читателя [1].

II. АЛГОРИТМЫ ИЗВЛЕЧЕНИЯ КЛЮЧЕВЫХ СЛОВ

1. TF-IDF

Для того чтобы оценить повторяемость слов в тексте, недостаточно просто подсчета слов, полученного от обычного счетчика. Это объясняется тем, что большое количества определенных общих слов в корпусе [2]. Это можно преодолеть при помощи векторизатора TF-IDF, который наказывает слова, повторяющиеся несколько раз в тексте. TF-IDF – это оценка частоты слов, которые выделяют слова более важные для контекста, а не те слова, которые просто часто появляются в документе.

2. RAKE

Rake (Rapid Automatic Keyword Extraction) – сравнительно эффективный алгоритм, который применяется для одиночных документов. Он может быть использован для анализа документов из различных предметных областей. Кроме того, этот алгоритм никак не зависит от структуры документа. Данный алгоритм основывается на том, что в ключевых словах редко встречаются шумовые слова и знаки препинания. Нужно также отметить, что такие слова как правило считаются не информативными и не участвуют в индексации в системах информационного поиска. Для работы алгоритма потребуется список таких слов, которые называются стоп-словами [3].

3. TextRank

TextRank – это алгоритм, основанный на PageRank, который часто используется для извлечения ключевых слов и суммаризации текста. PageRank – это алгоритм, используемый для расчета веса веб-страниц [4]. Например, можно представить все веб-страницы в виде большого направленного графа. В графе каждый узел – это веб-страница. Если веб-страница А имеет ссылку на веб-страницу Е, то эта связь может быть представлена в виде направленного ребра от А до Е.

4. Ансамбль методов

Можно использовать сразу несколько алгоритмов, а их результаты обрабатывать в соответствии с разработанными дополнительными алгоритмами.

Метод ансамбля будет состоять из следующих этапов:

- получение списка ключевых слов-кандидатов от каждого автоматического экстрактора ключевых слов;
- фильтрация списка ключевых слов-кандидатов;

- объединение и пересчет баллов ключевых слов-кандидатов;
- применение динамической пороговой функции для извлечения ключевых слов.

Общая схема для ансамбля двух алгоритмов представлена на рисунке 2.

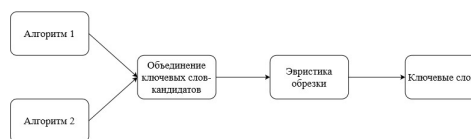


Рис. 2 – Общая схема ансамбля двух алгоритмов

III. ЗАКЛЮЧЕНИЕ

Понимание текста имеет своей целью проникновение в смысл текста. Смысл текста формируется на основе свернутой, явно невыраженной смысловой структуры, выводимой в соответствии с объемом лингвистических, энциклопедических, фоновых знаний человека, получающего эту информацию, его опыта и принадлежности к той или иной культурной общности. Потому предполагается множественность трактовок, возникающих в процессе восприятия и интерпретации. Эта структура представляет собой сжатое смысловое ядро текста, извлеченное из его ключевых слов, их ассоциативных связей и импликаций, которое должно быть сохранено при всех изменениях содержания интерпретаторами.

Существенной проблемой является и то, какие именно единицы текста являются решающими для понимания всего текста. Значение отдельных элементов текста для выражения общего смысла неодинаково, и наряду с центральными имеются также и второстепенные элементы текста. При изменении центральных элементов смысл текста меняется. Если изменить второстепенные единицы текста, то общий смысл может сохраниться. Актуальным на данный момент является вопрос о составлении объективной методики извлечения ключевых слов.

1. Методы выделения ключевых слов при реферировании научного текста [Электронный ресурс]. – Режим доступа: https://vestnik.tspu.edu.ru/files/vestnik/PDF/articles/moskvitina_t._n._45_50_8_197_2018.pdf.
2. TF-IDF [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/TF-IDF>.
3. Automatic Keyword Extraction from Individual Documents [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_-_d-from_Individual_Documents.
4. Understand TextRank for Keyword Extraction by Python [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/textrank-for-keyword-extraction-by-python-c0bae21bcec0>.

АЛГОРИТМ УЛУЧШЕНИЯ КАЧЕСТВА МЕДИАФАЙЛОВ

Минько Д. А.

Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: 98_minko@mail.ru

Рассматриваются основные подходы по улучшению качества медиафайлов. Обозревается подход на основе свёрточной нейронной сети, а также метод разреженного кодирования

ВВЕДЕНИЕ

Изначально многие медиафайлы сжаты с целью уменьшить размер файлов, что делает их идеальными для обмена и размещения в интернете. Вследствие этого, при попытке увеличить или использовать изображение повторно, оно может выглядеть зернистым или заретушированным. Повысить качество медиафайлов можно путем корректировки внешних атрибутов, цвета и контрастности изображения при помощи редактора изображений.

I. ОСНОВНЫЕ ПОДХОДЫ

Известно несколько основных подходов по улучшению изображения: предсказательные модели, краевые методы, статистические методы, методы основанные на паттернах, метод преобразования гистограмм, подход на основе свёрточной нейронной сети.

1. Предсказательное моделирование основывается на построении, управлении и расчете моделей при помощи техник аппроксимации;
2. Краевые методы - методы о нахождении решения заданного дифференциального уравнения (системы дифференциальных уравнений), удовлетворяющего краевым (граничным) условиям в концах интервала или на границе области;
3. Статистический метод – метод, основанный на анализе большой выборки данных (что и является статистикой);
4. Методы основанные на паттернах – методы, в основе которых лежат готовые паттерны (шаблоны) для решения конкретных задач.
5. Метод преобразования гистограмм. В данном методе пытаются достичь равномерности распределения яркостей обработанного изображения.
6. Свёрточная нейронная сеть - нейронная сеть, использующая множество идентичных копий одного и того же нейрона. Это позволяет сети иметь ограниченное число параметров при вычислении больших моделей. Свёрточная нейронная сеть, однажды обучив нейрон, использует его во множе-

стве мест, что облегчает обучение модели и минимизирует ошибки.

Наилучшее качество дают метод преобразования гистограмм и подход на основе свёрточной нейронной сети.

II. ОБЗОР ПОДХОДА НА ОСНОВЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Пусть у нас есть изображение Y , которое мы хотим восстановить ($F(Y)$), и оно должно быть максимально похожее на основное изображение высокого разрешения X . Данный подход состоит из трех операций:

1. Извлечение и представление патчей: эта операция извлекает (перекрывающиеся) патчи из изображения Y с низким разрешением и представляет каждый патч как вектор большой размерности.
2. Нелинейное отображение: эта операция нелинейно отображает каждый многомерный вектор на другой многомерный вектор. Каждый сопоставимый вектор концептуально представляет патч высокого разрешения. Эти векторы составляют другой набор карт характеристик.
3. Реконструкция: эта операция объединяет вышеприведенные патч-представления высокого разрешения для генерации окончательного изображения высокого разрешения. Ожидается, что это изображение будет похоже на X .

Первая операция выражается как:

$$F_1(Y) = \max(0, W_1 * Y + B_1), \quad (1)$$

Где W_1 и B_1 представляют фильтры и смещения соответственно, а «*» обозначает операцию свертки. На выходе после первой операции мы получим для каждого патча n_1 -мерный массив. Во второй операции происходит преобразование этого массива в n_2 -мерный, увеличивая размер первого массива для повышения точности. Каждый из выходных n_2 -мерных векторов концептуально представляет собой патч высокого разрешения, который будет использоваться для реконструкции:

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2). \quad (2)$$

В традиционных методах предсказанные перекрывающиеся участки с высоким разрешением часто усредняются для получения окончательного полного изображения. Усреднение можно рассматривать как предопределенный фильтр на наборе карт характеристик (где каждая позиция является «сплюсненной» векторной формой патча высокого разрешения). Основываясь на этом, мы определяем сверточный слой для получения окончательного изображения с высоким разрешением:

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3. \quad (3)$$

Хоть вышеописанные три операции интуитивно разные, но они все сводятся к той же форме, что и сверточный слой. Объединив их вместе, образуется сверточная нейронная сеть (рисунок 1). В этой модели все веса фильтрации должны быть оптимизированы.

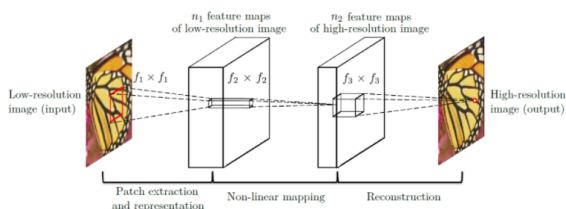


Рис. 1 – Визуализация работы сверточной нейронной сети

III. МЕТОД РАЗРЕЖЕННОГО КОДИРОВАНИЯ

Сверточная нейронная сеть имеет связь с методом разреженного кодирования (рисунок 2). Разреженное кодирование – это метод обучения представления, цель которого – поиск разреженного представления входных данных в форме линейных комбинаций базовых элементов, вместе с самими базовыми элементами. В методе разреженного кодирования предположим, что патч с низким разрешением $f_1 \times f_1$ извлекается из входного изображения. Затем патч проектируется в словарь (с низким разрешением). Если размер словаря равен n_1 , то это эквивалентно применению n_1 линейных фильтров на входное изображение. Это показано на левой части рисунка 3. Затем программа разреженного кодирования будет итеративно обрабатывать n_1 коэффициентов. Выходы этого решения – n_2 коэффициентов. Как правило, $n_1 = n_2$ в случае разреженного кодирования. Коэффициенты n_2 и представляют патч высокого разрешения. В этом случае программа разреженного кодирования ведёт себя как частный случай оператора нелинейного отображения. Это изображено на средней части рисунка 3. Вышеупомянутые коэффициен-

ты n_2 затем проектируются в другой словарь (с высоким разрешением) для создания фрагмента высокого разрешения. Перекрывающиеся участки усредняются, это эквивалентно линейным свёрткам на n_2 карты характеристик. На выходе получаем патчи с высоким разрешением размера $f_3 \times f_3$, тогда линейные фильтры имеют эквивалентную пространственную поддержку размера $f_3 \times f_3$.

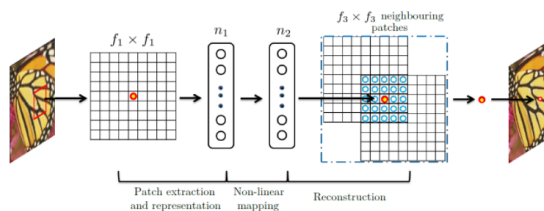


Рис. 2 – Связь сверточной нейронной сети с методом разреженного кодирования

Вышеупомянутая последовательность показывает, что метод на основе разреженного кодирования можно рассматривать как своего рода сверточную нейронную сеть (с другим нелинейным отображением). В данной сверточной сети словарь с низким разрешением, словарь с высоким разрешением. Нелинейное отображение вместе со средним вычитаемым усреднением задействованы в оптимизируемых фильтрах. Таким образом, этот метод оптимизирует сквозное отображение, состоящее из всех операций. Ниже приведена таблица, в которой показаны результирующие значения качества изображения и времени срабатывания, в зависимости от числа фильтров.

$n_1 = 128$ $n_2 = 64$		$n_1 = 64$ $n_2 = 32$		$n_1 = 32$ $n_2 = 16$	
PSNR	Time (sec)	PSNR	Time (sec)	PSNR	Time (sec)
32.60	0.60	32.52	0.18	32.26	0.05

IV. ВЫВОД

Улучшение качества медиафайлов – довольно важный элемент повседневной жизни и сферы технологий, который применяется буквально везде, где есть цифровое изображение. Данное направление ежедневно дорабатывается многими крупными компаниями, такими как Google, Apple. В результате проделанной работы было установлено, что подход на основе сверточной нейронной сети является одним из самых практичных и востребованных. Как минимум в сфере распознавания объектов или улучшения качества медиафайлов, или, проще говоря, везде, где необходимо обучать нейронную сеть. Такие результаты достигаются благодаря операциям свёртки.

АНАЛИЗ ЭТАПОВ ПРЕДОБРАБОТКИ И СЕГМЕНТАЦИИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ БОЛЬНЫХ COVID-19

Момотова Ю. О.

Факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: yuliya.momatava@gmail.com

В статье рассматриваются подходы, способные улучшить классификацию лёгких, поражённых новым коронавирусом 2019. Наиболее перспективными из рассматриваемых методов являются нелинейные усиления контраста изображения. В качестве исходных данных используются цифровые изображения рентгеновских снимков лёгких.

ВВЕДЕНИЕ

Новый коронавирус 2019 (COVID-2019), который впервые появился в китайском городе Ухань в декабре 2019 года, быстро распространился по миру и стал пандемией. Это оказало разрушительное воздействие как на повседневную жизнь, так и на здоровье населения и мировую экономику.

Для предотвращения стремительного распространения болезни необходимо как можно быстрее её диагностировать. Из-за этого потребность во вспомогательных диагностических инструментах возросла, так как на данный момент отсутствуют точные и автоматизированные способы обнаружения. Недавние результаты, полученные с использованием методов радиологической визуализации, предполагают, что такие изображения содержат важную информацию о вирусе COVID-19.

Применение передовых методов искусственного интеллекта в сочетании с радиологической визуализацией может быть полезным для точного выявления этого заболевания, а также может помочь преодолеть проблему нехватки специализированных врачей в отдалённых деревнях.

I. ОБЗОР ИСХОДНЫХ ДАННЫХ

Исходными данными являются цифровые изображения рентгеновских снимков лёгких людей. Они взяты из базы данных, размещённой в открытом доступе и содержащей реальные снимки [1].

Важным качеством цифрового изображения является возможность преобразования исходных данных для увеличения вероятности выделения потенциально информативных участков. Например, применение фильтров, наложение масок, изменение размеров и прочие трансформации исходного изображения. Это способствует нормализации изображения. Для этого непосредственно перед самым этапом распознавания с помощью нейронных сетей, выполняются этапы предобработки и сегментации. На этапе предобработки осуществляется улучшение качество изображения, а на этапе сегментации — вы-

деление только требуемой для анализа области и отсечение всего лишнего.

На данный момент уже существует ряд исследований и научных статей, описывающих различные реализации автоматического распознавания заболевания на основе рентгеновских снимков [2–6]. Однако в них подробно не рассмотрены вопросы предобработки и сегментации изображений. Для улучшения результатов распознавания стоит рассмотреть внедрение полноценного этапа предобработки и сегментации.

Основные проблемы, решением которых занимаются данные этапы:

- искажённая яркостная характеристика;
- низкая контрастность;
- наличие небольшого шума;
- небольшие колебания в расположении лёгких, обусловленные физиологией и условиями создания снимка;
- отсечение лишней информации, которая не участвует в анализе.

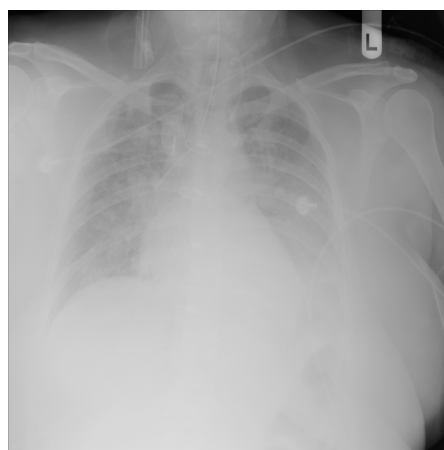


Рис. 1 – Пример рентгеновского снимка лёгких

Как видно на приведённом примере (см. рис. 1), область лёгких нечёткая и малоконтрастная, присутствуют шумы и имеется небольшое смещение. Также для анализа необходима только область лёгких, а плечи, руки, шея и область живота излишние. Более того, наличие этих областей при обучении нейронных сетей может привести к искажению результатов.

II. МЕТОДЫ УЛУЧШЕНИЯ КОНТРАСТА ИЗОБРАЖЕНИЯ

Контрастностью является разницей между максимальной и минимальной яркостью пикселей на изображении. И эта разница может быть как глобальная, так и локальная, в зависимости от области, на которой требуется найти данные значения.

Под резкостью принято понимать отношение разности яркостей соседних фрагментов, т.е. их контрастность, к ширине контурной линии, разделяющей эти фрагменты, т.е. фактически пространственную производную яркости. Очевидно, что в отличие от контрастности резкость может быть только локальной. Исходя из данного определения резкость можно повысить, или сократив ширину контурной линии, или увеличив контрастность фрагментов. Причем повышение контрастности всегда приводит к увеличению резкости.

В используемом наборе данных изображения зачастую являются малоконтрастными. Слабый контраст, как правило, обусловлен широким диапазоном воспроизводимых яркостей, нередко сочетающийся с нелинейностью характеристики передачи уровней. Характер зависимости изменения яркости палитры пикселей от минимального значения до максимального также влияет на качество изображения. Оптимальной является линейная функция изменения интенсивности пикселей. При вогнутой характеристике изображение будет более темным, при выпуклой — более светлым. И в том, и в другом случае признаки объектов могут быть искажены и недостаточно хорошо идентифицируемы. Коррекция яркости палитры существенно улучшает качество изображения.

Методы увеличения контрастности и, как следствие, увеличения резкости можно разделить на две большие группы:

- линейные, в основе которых лежит преобразование глобальной шкалы яркости;
- нелинейные, в которых величина изменения яркости каждого пикселя зависит от характеристики окружающей его окрестности.

Линейные алгоритмы учитывают только глобальные характеристики изображения. Они являются оптимальными при гауссовском распределении сигналов, помех и наблюдаемых данных. Однако реальные изображения не подчиняются данному распределению вероятностей (разнообразные перепады яркости на границах, переходы от одной текстуры к другой и прочее). В данном случае, при использовании рентгеновских снимков, существуют проблемы, т.к. присутствует очень сильный контраст между различными областями.

Нелинейные алгоритмы в свою очередь рассматривают значения основываясь на данных

определённой области для каждого конкретного пикселя. Это решает проблему наличия слишком контрастных областей, так как для них будут рассчитываться параметры отдельно, что помогает не потерять важную информацию.

Таким образом в случае рентгеновских изображений, на которых присутствуют сильные перепады областей чёрного и белого цвета, для увеличения контраста локальных областей необходимо использовать нелинейный алгоритм.

ЗАКЛЮЧЕНИЕ

В данной статье описаны базовые проблемы изображений, которые используются для классификации больных COVID-19 на основе рентгеновских снимков лёгких. Анализ существующих решений показал, что классификация происходит без каких либо этапов предобработки исходных данных. Основной проблемой таких данных является низкая контрастность, шум, избыточность.

Как вероятное эффективное решение в данной статье предлагается введение этапов предобработки и сегментации перед этапом классификации. Это будет способствовать нормализации данных, поступающих в классификатор.

Для этапа предобработки предлагается использовать один из нелинейных алгоритмов повышения контрастности. И дальнейшим направлением этой статьи является изучение и сравнение таких алгоритмов для нахождения оптимального.

1. X-Ray Image DataSet [Electronic resource] – Mode of access: <https://github.com/muhammedtalo/COVID-19/tree/master/X-Ray%20Image%20DataSet>. – Date of access: 23.10.2020.
2. Automated detection of COVID-19 cases using deep neural networks with X-ray images [Electronic resource] / T. Ozturk, M. Talo, E. A/ Yildirim. – Mode of access: <https://doi.org/10.1016/j.compbimed.2020.103792>. – Date of access: 20.10.2020.
3. Detecting COVID-19 in X-ray images with Keras, TensorFlow, and Deep Learning [Electronic resource] / A. Rosebrock. – Mode of access: <https://www.pyimagesearch.com/2020/03/16/detecting-covid-19-in-x-ray-images-with-keras-tensorflow-and-deep-learning/>. – Date of access: 21.10.2020.
4. COVID-19 X-Ray Image Classification with mxnet [Electronic resource] / A. Galst. – Mode of access: <https://medium.com/analytics-vidhya/covid-19-xray-image-classification-with-mxnet-f09c752a8e67>. – Date of access: 21.10.2020.
5. Deep learning approaches for COVID-19 detection based on chest X-ray images [Electronic resource] / A. M. Ismael – Mode of access: <https://doi.org/10.1016/j.eswa.2020.114054>. – Date of access: 20.10.2020.
6. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization [Electronic resource] / T. Mahmud, A. Rahman, A. Fattah. – Mode of access: <https://doi.org/10.1016/j.compbimed.2020.103869>. – Date of access: 21.10.2020.

МЕТОДЫ МАТРИЧНОЙ ФАКТОРИЗАЦИИ ДЛЯ СИСТЕМ РЕКОМЕНДАЦИИ

Мойсюк-Дранько П. А., Ревотюк М. П.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: moysyukdrankopasha@gmail.com, rmp@bsuir.by

Рассматриваются общие технические сведения рекомендательных систем и алгоритма матричной факторизации

ВВЕДЕНИЕ

Современные потребители забиты выбором. Электронные ритейлеры и контент-провайдеры предлагают огромный выбор продуктов, с беспрецедентными возможностями для удовлетворения разнообразных особых потребностей и вкусов. Подбор наиболее подходящих продуктов для потребителей является ключом к повышению удовлетворенности пользователей и лояльности. Поэтому все больше розничных продавцов проявляют интерес к системам рекомендаций, которые анализируют шаблоны интереса пользователей к продуктам, чтобы дать персонализированные рекомендации, соответствующие вкусу пользователя. Поскольку хорошие персонализированные рекомендации могут добавить еще одно измерение к опыту пользователя, лидеры электронной коммерции, такие как Amazon.com и Netflix, сделали системы рекомендаций важной частью своих вебсайтов.

I. ОБЩИЕ СВЕДЕНИЯ

В широком смысле, системы рекомендаций основаны на одной из двух стратегий. Подход контент-фильтрации создает профиль для каждого пользователя или продукта, чтобы охарактеризовать его природу. Профили позволяют программам ассоциировать пользователей с соответствующими продуктами. Конечно, контент-ориентированные стратегии требуют сбора внешней информации, которая может быть недоступна или которую трудно собрать. Альтернатива контент-фильтрации основывается только на прошлом поведении пользователя, например, на предыдущих транзакциях или рейтингах продуктов, без необходимости создания явных профилей. Этот подход известен как коллаборативная фильтрация. Коллаборативная фильтрация анализирует отношения между пользователями и взаимозависимости между продуктами для выявления новых ассоциаций между пользователями. Двумя основными областями коллаборативной фильтрации являются методы соседства и модели скрытых факторов. Методы соседства сосредоточены на вычислении отношений между элементами или, в качестве альтернативы, между поль-

зователями. Элементно-ориентированный подход оценивает предпочтения пользователя по отношению к элементу на основе оценок "соседей" элементов одним и тем же пользователем. Соседями продукта являются другие продукты, которые, как правило, получают аналогичные рейтинги, если их оценивает один и тот же пользователь. Некоторые из наиболее успешных реализаций латентных факторных моделей основаны на матричной факторизации. В своей базовой форме матричная факторизация характеризует как элементы, так и пользователей по векторам факторов, выведенных из рейтинговых моделей элементов. Высокое соответствие между элементным и пользовательским факторами приводит к рекомендации. Эти методы стали популярными в последние годы благодаря сочетанию хорошей масштабируемости и точности прогнозирования. Кроме того, они предлагают большую гибкость для моделирования различных реальных ситуаций. Рекомендующие системы полагаются на различные типы входных данных, которые часто помещаются в матрицу с одним измерением, представляющим пользователей, и другим измерением, представляющим элементы, представляющие интерес. Наиболее удобными данными являются высококачественные явные обратные связи, которые включают в себя явный ввод данных пользователями относительно их интереса к продуктам. Одним из достоинств матричной факторизации является то, что она позволяет включать дополнительную информацию. Когда явная обратная связь недоступна, системы референтов могут делать выводы о предпочтениях пользователя, используя неявную обратную связь, которая косвенно отражает мнение, наблюдая за поведением пользователя.

II. БАЗОВАЯ МАТРИЧНАЯ МОДЕЛЬ ФАКТОРИЗАЦИИ

Матричные факторизационные модели отображают как пользователей, так и элементы в единое латентное факторное пространство размерности f таким образом, что взаимодействия между пользователем и элементами моделируются как внутренние продукты в этом

пространстве. Соответственно, каждый элемент i ассоциируется с вектором $q_i \in \mathbb{R}^f$ и каждый пользователь u связан с вектором $p_u \in \mathbb{R}^f$

Для данного пункта i , элементы q_i измеряют степень, в которой данный пункт обладает этими факторами, положительными или отрицательными. Для данного пользователя u , элементы p_u измеряют степень интереса, который пользователь имеет к элементам, высоким по соответствующим факторам, опять же, положительным или отрицательным. Получившийся точечный продукт, $q_i^T p_u$ фиксирует взаимодействие между пользователем u и объектом i - общий интерес пользователя к характеристикам объекта. Это приблизительно соответствует рейтингу пользователя u пункта i , который обозначается r_{ui} , что приводит к оценке

$$\hat{r}_{ui} = q_i^T p_u \quad (1)$$

Главная задача заключается в вычислении соотношения между каждым элементом и пользователем с учетом векторов $q_i, p_u \in \mathbb{R}^f$. После того, как рекомендующая система выполнит это отображение, она может легко оценить рейтинг, который пользователь присвоит любому элементу с помощью уравнения 1.

Такая модель тесно связана с сингулярной величиной декомпозиции (SVD), хорошо зарекомендовавшей себя методикой выявления скрытых семантических факторов при поиске информации. Применение SVD в области совместной фильтрации требует факторинга матрицы рейтинга пользовательского элемента. Это часто вызывает трудности из-за высокой доли пропущенных значений, вызванной редкостью в матрице рейтингов пользовательского элемента. Обычное SVD не определено, когда знания о матрице неполные. Более того, неосторожное обращение только к относительно небольшому количеству известных элементов очень склонно к переподгонке.

Ранние системы полагались на приписывание, чтобы заполнить недостающие рейтинги и сделать матрицу рейтинга плотной [2]. Однако приписывание может быть очень дорогостоящим, так как оно значительно увеличивает объем данных. Кроме того, неточные вменения могут значительно исказить данные. Поэтому в более поздних работах [3-6] предлагалось моделировать только непосредственно наблюдаемые рейтинги, избегая при этом переполнения через упорядоченную модель. Для изучения факторных векторов (p_u и q_i) система минимизирует ошибку регуляризованного квадрата на множестве известных рейтингов:

$$\min_{q, p} \sum_{(u, i) \in \mathbb{K}} (r_{ui} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2) \quad (2)$$

Здесь K - это набор пар (u, i) , по которым известен r_{ui} (тренировочный набор).

Система знакомится с моделью, подгоняя ее под ранее наблюдавшиеся рейтинги. Однако цель состоит в том, чтобы обобщить эти предыдущие рейтинги таким образом, чтобы предсказать будущие, неизвестные рейтинги. Таким образом, система должна избегать переподгонки наблюдаемых данных путем регуляризации выученных параметров, величины которых оштрафованы. Константа λ контролирует степень регуляризации и обычно определяется перекрестной проверкой. Вероятностным основанием для регуляризации является "Вероятностный матричный фактор"[7] Руслана Салахутдинова и Андрея Мниха.

III. АЛГОРИТМЫ ОБУЧЕНИЯ

Два подхода к минимизации уравнения 2 - это стохастический градиентный спуск и чередующиеся наименьшие квадраты (ALS).

Вывод

Методы матричной факторизации стали доминирующей методикой среди рекомендующих совместную фильтрацию. Опыт работы с наборами данных, такими как данные Netflix Prize, показал, что они обеспечивают точность, превосходящую классические методы "ближайшего соседа". В то же время, они предлагают компактную модель с эффективной памятью, которую системы могут относительно легко изучить. Еще более удобным является то, что модели могут естественным образом интегрировать многие важные аспекты данных, такие как множественные формы обратной связи, временная динамика и уровни уверенности.

СПИСОК ЛИТЕРАТУРЫ

1. Goldberg D., Using Collaborative Filtering to Weave an Information Tapestry / D. Goldberg // Comm. ACM. - 1992. - Vol. 32, pp. 61-70.
2. Application of Dimensionality Reduction in Recommender System - A Case Study / B. M. Sarwar // ACM Press. - 2000
3. Netflix update: Try This at Home [Electronic resource] / Funk S., Dec. 2006. Mode of access: <http://sifter.org/~simon/journal/20061211.html>. - Date of access: 03.08.2020.
4. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model / Y. Koren // ACM Press. -2008. -Proc. 14 ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining. - pp. 426-434.
5. Improving Regularized Singular Value Decomposition for Collaborative Filtering / A. Paterek // ACM Press. - 2007. -Proc. KDD Cup and Workshop. -pp. 39-42
6. Major Components of the Gravity Recommendation System / G. Takacs et al. // SIGKDD Explorations, vol.9. -2007. - pp. 80-84
7. Probabilistic Matrix Factorization / R. Salakhutdinov and A. Mnih // ACM Press. -2008. -Proc. Advances in Neural Information Processing Systems 20 (NIPS 07). - pp. 1257-1264 <http://www.inform.ind.edu/PBIO/>. - Date of access: 14.09.2012.

FLAT PROBLEM OF ALLOWANCE DISTRIBUTION AS DUAL CONTROL PROBLEM

Mukha V. S., Kako N. F.

Department of Information Technologies of Automated Systems, Belarusian State University of Informatics and Radioelectronics

Minsk, Republic of Belarus

E-mail: mukha@bsuir.by, kako.nancy@gmail.com

The problem of the allowance distribution is described. This problem is formulated as a dual control problem.

INTRODUCTION

The allowance is the layer of the material that is removed from the workpiece (billet) in the machining or finishing. The example of the flat workpiece (solid line) and the template (dotted line) is showed in Fig. 1. The difference between these two curves forms the allowance that should be cut from this workpiece. The allowance is shaded in Fig. 1. The allowance distribution is the process of the placing of the template on the workpiece in the best possible way [1]. If the allowance is too small, then it is possible to spoil the template in the machining. If the allowance is too big, then the big part of material will go in chips, and the cost of processing will increase. Therefore, the production is always trying to make the allowance as uniform as possible. The allowance distribution problem is a complex problem, the various aspects of which are considered in works [2-5].

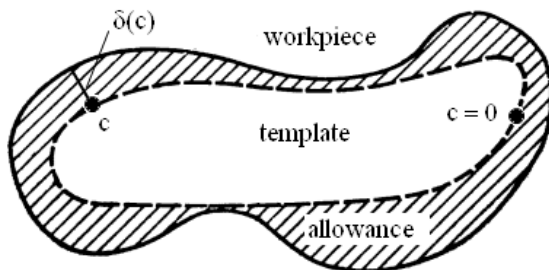


Fig. 1 – The workpiece and the template in the problem of the allowance distribution

I. MATHEMATICAL MODEL OF THE CONTROLLED OBJECT

In this report, we consider the flat problem of the allowance distribution. The value of the allowance is measured by the normal to the curve characterizing the shape of the template. Then the allowance is some function of the position: $\delta = \delta(c)$, where c is the curvilinear coordinate directed along the contour of the template, $0 \leq c \leq c_0$, c_0 is the perimeter of the contour (Fig. 1). The function δ depends on the position of the template's contour on the workpiece. The position of the template's contour is characterized in the flat case by three coordinates. They may be, for instant, the abscissa x , ordinate y of the mass center and the rotation

angle ϕ relative the axis x (or basis direction): $\delta = \delta(c, x, y, \phi)$.

The non-uniformity of the allowance could be defined, for instance, as follows [1]:

$$Q = Q(x, y, \phi) = \max_c \delta(c, x, y, \phi) - \min_c \delta(c, x, y, \phi), \quad (1)$$

i.e. as the difference between the maximum and minimum values of the allowance (Fig. 2).

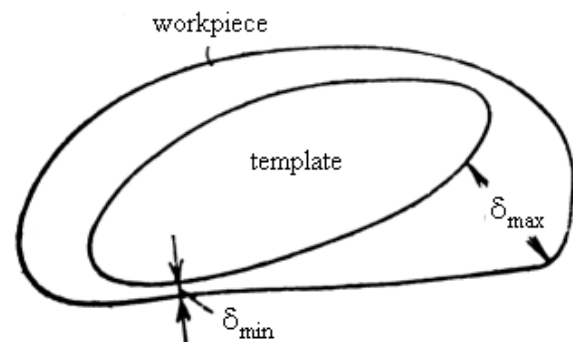


Fig. 2 – The relation between the workpiece and the template in the problem of the allowance distribution, $\delta_{max} = \max_c \delta(c, x, y, \phi)$, $\delta_{min} = \min_c \delta(c, x, y, \phi)$

The problem of the optimal allowance distribution can be formulated as follow: it is needed to minimize the function $Q(x, y, \phi)$ (1) on the variables x, y, ϕ :

$$Q(x, y, \phi) \rightarrow \min_{x, y, \phi}.$$

The function (1) is unknown beforehand and could be estimated on the base of measurements. Measurements could be organized as follows. The values of the allowance $\delta = \delta(c, x, y, \phi)$ are determined practically in the m_t points p_1, p_2, \dots, p_{m_t} on the contour of the template by the sensors. As a results, they receive the values $\delta_i = \delta(p_i, x, y, \phi)$, $i = 1, 2, \dots, m_t$, when x, y, ϕ are fixed.

Thus, the allowance distribution object is described practically by the following function:

$$Q = Q(x, y, \phi) = \max(\delta_1, \dots, \delta_{m_t}) - \min(\delta_1, \dots, \delta_{m_t}). \quad (2)$$

The measurements $\delta_1, \dots, \delta_{m_t}$ give one value of the output variable Q .

The contour of the workpiece is defined practically also by its points r_1, \dots, r_{m_w} . The

measurements Q under fixed x, y, ϕ contain errors due to the point-defined outlines of the workpiece and the template and are random. This means that the object with description (2) is a regression object and is unknown beforehand.

II. DUAL CONTROL OF THE ALLOWANCE DISTRIBUTION

The allowance distribution problem can be considered as the problem of dual control of the regression object. The dual control is recommended in automatic search systems and in particular automatic optimization systems [6]. The dual control allows to store the information about the object in the control process. The control actions must have a reciprocal, “dual” character: to a certain extent they must be of a probing nature, but also controlling to a known degree.

Fig. 3 shows the block diagram of the proposed closed-loop automatic control system for the optimal (dual) allowance distribution. We will suppose that the system depicted in Fig. 3 is discrete-continuous. It means that all the variables in the diagram are considered only at the discrete time instants $s = 0, 1, 2, \dots, n$. Any of the variables at the time instant s is provided with a subscript s , for example, $Q_s, u_s = (x_s, y_s, \phi_s)$. We have the measurements $Q_s = Q(x_s, y_s, \phi_s)$, $s = 1, 2, \dots, n$, where n is the amount of the time instants (the volume of the sample), $Q_s, u_s = (x_s, y_s, \phi_s)$ are the output and input variables of the controlled object respectively.

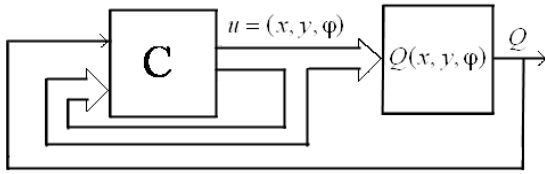


Fig. 3 – The block diagram of the closed-loop automatic control system for the optimal (dual) allowance distribution

Let $f_{Q_s}(q_s/x_s, y_s, \phi_s, \Theta) = f_{Q_s}(q_s/u_s, \Theta)$ be the known conditional probability density function of the random variable Q_s , where Θ is the set of the unknown parameters of the probability density function. We will suppose that the parameters Θ are random and their probability density function $f_{\Theta}(\theta)$ is known. The control vector u_s is restricted by the condition $u_s \in U$, where U is a certain admissible region, for example, $U = \{u_s : u_s \geq 0, \psi_j(u_s) \leq 0, s = 0, 1, \dots, n, j = 1, 2, \dots, l\}$, and $\psi_j(u_s)$ are some scalar functions of the vector u_s . It is assumed, that the controller C utilizes past information in the form of the measurements $\vec{u}_{s-1} = (u_0, u_1, \dots, u_{s-1})$, $\vec{Q}_{s-1} = (Q_0, Q_1, \dots, Q_{s-1})$ to form the control action u_s at the time instant s . It is assumed, that the controller C has a random behavior and is described at the time instant s by the conditional probability density function $\Gamma_s = f(u_s/\vec{u}_{s-1}, \vec{Q}_{s-1})$ called the strategy of the controller. The quality

of the functioning of the system at the time instant s is evaluated by the loss function $W_s(Q_s)$. Mathematical expectation of the loss function $R_s = E\{W_s(Q_s)\}$ is called mean specific risk. The sum of the mean specific risks on the $n + 1$ time instants

$$R = E\left\{\sum_{s=0}^n W_s\right\} = \sum_{s=0}^n E\{W_s\} = \sum_{s=0}^n R_s \quad (3)$$

is called the mean total risk. The system will be optimal when the mean total risk R will be minimal. The problem consists of determining the sequence of the strategies of the controller $\Gamma_s = f(u_s/\vec{u}_{s-1}, \vec{Q}_{s-1})$, $s = 0, 1, 2, \dots, n$, minimizing the mean total risk R (3).

The strategies of the controller $\Gamma_s = f(u_s/\vec{u}_{s-1}, \vec{Q}_{s-1})$ minimizing the mean total risk R (3) are not randomized, i.e. $\Gamma_s = \delta(u_s - u_s^*)$, δ is the unit impulse function, u_s^* is optimal control action [6]. The sequence of the control actions $u_n^*, u_{n-1}^*, \dots, u_0^*$ is defined from the functional equations which can be found in the work [7].

We will suppose also that the probability density function $f_{Q_s}(q_s/u_s, \Theta)$ is normal $N(\psi(u_s, \Theta), \sigma^2)$, where $\psi(u_s, \Theta)$ is the regression function and σ^2 is constant. We propose to use the quadratic regression function

$$\psi(u_s, \Theta) = \theta_0 + \theta_1 u_s + \theta_2 u_s^2, \quad \Theta = \{\theta_0, \theta_1, \theta_2\},$$

where θ_0 is constant, θ_k is the k -dimensional matrix of the order 3, ${}^{0,k}(\theta_k u_s^k)$, $k = 1, 2$, is the $(0, k)$ -folded product of the multidimensional matrices [8]. The loss function is proposed to be $W_s = Q_s$.

III. REFERENCES

1. Rastrigin L.A. Systems of extremal control. Moscow, Nauka, 1974. 632 p. In Russian.
2. Lapteva E.N. Automated system of large-size complex profile hardware allowance distribution technological process control (in screw-propellers manufacture): auto abstract of the thesis to the PhD degree of the technical sciences. Moscow, 2004. 16 p. In Russian.
3. Lapteva E.N., Rogov V.A. Automated system of allowance distribution technological process control. The problems of ship engineering. Severodvinsk, 2004. Issue 3. Pp. 52–57. In Russian.
4. Ying Zhang, Dinghua Zhang, Baohai Wu. An approach for machining allowance optimization of complex parts with integrated structure. Journal of Computational Design and Engineering 2 (2015) 248–252.
5. Yu-Wen Sun, Jin-Ting Xu, Dong-Ming Guo, Zhen-Yuan Jia. A unified localization approach for machining allowance optimization of complex curved surface. Precision Engineering. 2009. 33 (4), 516–523.
6. Feldbaum A.A. Fundamentals of the theory of the optimal automatic systems. Moscow, Nauka, 1963. 553 p. In Russian
7. Mukha V. S., Kako N. F. Dual Control of Multidimensional-matrix Stochastic Objects // Information Technologies and Systems 2019 (ITS 2019): Proceeding of the International Conference, BSUIR, Minsk, 30th October 2019. Minsk: BSUIR, 2019. Pp. 236–237.
8. V.S. Mukha. Analysis of multidimensional data. Minsk, Technoprint, 2004. 368 p. In Russian.

АНАЛИЗ МИРОВОЙ ЭКОНОМИКИ НА БАЗЕ ТАБЛИЦ «ЗАТРАТЫ – ВЫПУСК» WIOD И EORA СРЕДСТВАМИ ЯЗЫКА R

Пархименко В. А., Быков А. А.

Белорусский государственный университет информатики и радиоэлектроники

Белорусский государственный экономический университет

Минск, Республика Беларусь

E-mail: parkhimenko@bsuir.by, aliaksei.bykau@yandex.ru

В публикации дается краткое описание содержательных и технических характеристик таблиц «затраты – выпуск», публикуемых в рамках проектов WIOD и Eora. Приводится обзор некоторых задач, которые можно ставить и решать на основе указанных данных средствами языка R. Рассматриваются проблемы использования данных WIOD и Eora для макроэкономического анализа белорусской экономики.

Межрегиональная таблица «затраты – выпуск» (Multi-regional Input-Output table, MRIO) – это таблица, построенная по стандартной методологии «затраты – выпуск», однако учитывается информация не только по отраслям, но и в территориальном разрезе – по нескольким регионам. Она состоит из трех квадрантов (промежуточное потребление, конечное потребление и добавленная стоимость) и может быть описана системой линейных уравнений, решение которых хорошо описано в литературе.

О межрегиональных таблицах писал еще создатель методологии «затраты – выпуск» В. В. Леонтьев в 1963 году [1]. Он же в 1977 году опубликовал работу, в которой предложил описать мировую экономику средствами указанной методологии, выделив всего два региона – развитые и развивающиеся страны [2]. В случае проектов WIOD и EORA речь идет именно о такой модели – модели «затраты – выпуск» для всей мировой экономики как единого целого, т.е. регионами в данном случае выступают отдельные страны.

Схематично межрегиональная таблица «затраты – выпуск» представлена на рис. 1.

		Intermediate use		Final demand		Gross output
		Country A Industry	Country B Industry	Country A Industry	Country B Industry	
Country A	Industry	Intermediate use of domestic output	Intermediate use by B of exports from A	Final use of domestic output	Final use by B of exports from A	X_A
Country B	Industry	Intermediate use by A of exports from B	Intermediate use of domestic output	Final use by A of exports from B	Final use of domestic output	X_B
Value added		V_A	V_B			
Gross input		X_A	X_B			

Exports from A to B of intermediates Exports from A to B of final products

Рис. 1 – Схематичная структура межрегиональной таблицы «затраты – выпуск» [3]

Данные WIOD содержат информацию о 43 странах (26 стран ЕС и 15 крупных экономик мира), а также 44-ю квази-страну под названием «Весь остальной мир» (rest of the world).

По каждой стране имеется 56 секторов экономики. Помимо традиционных трех квадрантов таблицы «затраты – выпуск», WIOD содержит 16 социо-экономических показателей (Socio Economic Accounts) и 5 показателей экологического характера (Environmental Accounts).

Последний выпуск WIOD – ноябрь 2016 года. В феврале 2018 года были обновлены значения показателей экологического характера. Полные таблицы «затраты – выпуск» представлены и свободно доступны для скачивания за 2000-2014 годы. Официальный сайт проекта WIOD – wiod.org, а подробное описание методологии можно прочитать в [4].

Данные Eora являются более детализированными, так как охватывают 190 стран, правда, используемая классификация секторов экономики более узкая – 26 секторов для версии Eora26. Что касается дополнительных показателей, то в Eora содержится 2720 экологических и социо-экономических показателей.

Eora в отличие от WIOD охватывает и более длительный период: данные представлены за 1990-2015 годы (в том числе учтена информация по СССР). При этом доступны как страновые данные (Individual country IO tables), так и полная версия (Full Eora) и ее упрощенный вариант – Eora26. Официальный сайт проекта – worldmrio.com, а описание представлено в [5, 6].

Данные WIOD доступны в формате Excel, RData и STATA, а данные Eora доступны в текстовом формате (tab-separated .txt files). При этом в Eora квадранты основной таблицы, метки (labels) данных и дополнительные показатели располагаются в отдельных файлах.

Таблица WIOD имеет размерность 2690 x 2472, т.е. 6,6 млн. элементов, а размерность только первого квадранта (матрица транзакций в рамках промежуточного потребления) в Eora составляет 4915 x 4915, т.е. почти 25 млн. элементов.

Предпочтительный инструмент работы с Eora не Excel, а MATLAB, Stata, Octav, R или Python. Данные WIOD менее требовательны к

инструментам анализа и мощности вычислительной техники, однако выбор Excel не выглядит обоснованным.

В качестве основных задач, которые можно решать на основе данных WIOD и Eora, стоит упомянуть прогнозирование выпуска в рамках мировой экономики (по отдельной стране или сектору), оценку устойчивости технологических коэффициентов на протяжении последних десятилетий, оценку экологической нагрузки (environmental footprints) по, например, выбросам CO_2 , оценку социальных последствий – например, изменения уровня занятости, анализ глобальных цепочек добавленной стоимости (global value chains), изучение одной конкретной отрасли в рамках глобальной экономики [7] или прогнозирование влияния COVID-19 на валовой выпуск и ВВП (см. методику такой оценки в [8]).

В качестве примера приведем часть скрипта на языке R, который на основе данных WIOD осуществляет получение матрицы (2464 x 2464) с информацией о торговле добавленной стоимостью по каждой стране и каждому сектору:

```
# Calculation of technological coefficients
for (i in 1:2464){
  if(Total_output_matrix[i]!=0){
    A[,i] <- A[,i] / Total_output_matrix[i]}
  else A[,i] <- 0
}

# Creating value added coefficients matrix
Value_added <- diag(rep(1,2464)-colSums(A))

# Finding Leontief inverse matrix
B <- solve(diag(1,2464) - A)

# Creating export coefficients matrix
Gross_export <- Total_output_matrix
for (i in 1:44){
  for (j in 1:56){
    Gross_export[56*(i-1)+j] <-
    Gross_export[56*(i-1)+j] -
    sum(Transactions_matrix[56*(i-1)+j,
      (56*(i-1)+1):(56*(i-1)+56)]) -
    sum(Final_demand_matrix[56*(i-1)+j,
      (5*(i-1)+1):(5*(i-1)+5)])
  }
}
Gross_export <- diag(Gross_export)

# Calculating trade in value added
TVA <- Value_added %*% B %*% Gross_export
```

Отметим, что работа с данными WIOD и Eora сопряжена с рядом сложностей. Далеко не по всем странам есть полная и качественная статистика. Это, хотя и не мешает работать с общей моделью глобальной экономики,

так как основные страны-субъекты мировой экономики представлены хорошо, все же не позволяет проводить глубокий анализ отдельных деталей общей картины. Известно также, что официальные данные разных стран по взаимному импорту-экспорту противоречат друг другу, поэтому WIOD и Eora отражают некоторые «компромиссные» решения. Кроме того, в представленных данных есть опечатки и некорректные значения (нулевые, отрицательные). Несбалансированность таблицы (Eora) или ее искусственная балансировка (WIOD) – еще одна проблема. Наконец, как указывалось ранее, последние актуальные данные относятся к 2014–2015 годам.

Чрезвычайно привлекательно использовать WIOD и Eora для исследования белорусской экономики, степени ее вовлеченности в глобальные технологические цепочки, подверженности различным «шокам» и структурным сдвигам, происходящим в других странах. Однако данные по Беларуси отсутствуют в WIOD (Беларусь неявно учтена в Rest of the World, но это ничего не дает с практической точки зрения), а в Eora данные по Беларуси неполные и неточные.

1. Leontief, W. Multiregional input-output analysis / W. Leontief, A. Strout // Structural interdependence and economic development. – Palgrave Macmillan, London, 1963. – P. 119–150.
2. Leontief, W. The future of the world economy / W. Leontief // Socio-Economic Planning Sciences. – Vol. 11. – № 3. – P. 171–182.
3. UNCTAD. Global value chains and development: Investment and value added trade in the global economy. – 2013. – Mode of access: https://unctad.org/system/files/official-document/diae2013d1_en.pdf. – Date of access: 24.10.2020.
4. Timmer, M. P. An Illustrated User Guide to the World Input–Output Database: the Case of Global Automotive Production / M. P. Timmer, E. Dietzenbacher, B. Los, R. Stehrer, G. J. de Vries // Review of International Economics. – 2015. – Vol. 23. – №3. – P. 575–605.
5. Lenzen, M. Mapping the structure of the world economy / M. Lenzen, K. Kanemoto, D. Moran, A. Geschke // Environmental Science and Technology. – 2012. – Vol.46. – №15. – P. 8374–8381.
6. Lenzen, M. Building Eora: A Global Multi-regional Input-Output Database at High Country and Sector Resolution / M. Lenzen, K. Kanemoto, D. Moran, A. Geschke // Economic Systems Research. – Vol. 25. – № 1. – P. 20–49.
7. Пархименко, В.А. Анализ мировой ИТ-индустрии на базе таблиц «затраты- выпуск» средствами языка R / В. А. Пархименко // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: VI Междунар. науч.-практ. конф., Минск, 20-21 мая 2020 года: в 3 ч. Ч. 2. – Минск : Бестпринт, 2020. – С. 376–386.
8. Быков, А. А. Влияние COVID-19 на российскую экономику: методологические подходы к оценке на основе межотраслевого баланса / А. А. Быков, В. А. Пархименко, С. А. Толкачев // Белорусский экономический журнал. – 2020. – №2. – С. 25–37.

ВСТРЕЧНЫЙ ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ДИНАМИЧЕСКИ ОПРЕДЕЛЯЕМЫХ ГРАФАХ

Ревотюк М. П., Хаджинова Н. В., Кузнецова О. В.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: {rmp, kafitas}@bsuir.by

I. ПОСТАНОВКА ЗАДАЧИ

Известно, что для поиска кратчайших путей на нагруженном ориентированном графе $G(N, A)$, где N – множество вершин, A – множество дуг с весовой функцией $W : A \rightarrow R^+$, лучшим является алгоритм Дейкстры [1]. Процесс построения дерева кратчайших путей имеет волновой характер до исчерпания возможности его развития из исходной вершины. В этом случае для каждой вершины x требуется представление множества непосредственно достижимых смежных вершин x' , $x' = \{y | w(x, y) \geq 0\}$, где $w(x, y)$ – вес дуги $x \rightarrow y, y \in N$.

Пространство состояний поиска решения алгоритмом Дейкстры традиционно включает: D – массив расстояний от корня дерева, $|D| = |N|$; P – массив номеров предшествующих вершин, $|P| = |N|$; Q – очередь вершин, $|Q| \leq |N|$, где элементы упорядочены по текущему значению расстояния от корня дерева [1,2].

На практике возникает потребность поиска путей на графах с изменяемой структурой, когда описание графа задана легко модифицируемым списком дуг. Реляционная модель графа и пространства поиска решения – основа эффективной реализации поиска дерева путей методом бутстрэппинга [2]. Такой метод отличается ограничением потребности в памяти количеством фактически просматриваемых вершин без оценки предельных значений $|N|$, а также возможностью реализации открытых для расширения систем.

Предмет обсуждения – развитие и обобщение приемов реализации метода бутстрэппинга [2] в задачах встречного поиска пути между парами вершин графа с переменной структурой.

II. МОДЕЛЬ ПРОСТРАНСТВА СОСТОЯНИЯ ПОИСКА

Дерево кратчайших путей – связный граф по определению. Если s – исходная вершина, а x – произвольный узел или лист дерева путей, $s, x \in N$, то после завершения поиска последовательность посещаемых вершин обратного пути

$$p(x, s) = \{x, P(x), P(P(x)), \dots, s\} \quad (1)$$

после тривиального изменения направления перечисления элементов представляет искомым кратчайший путь. Элементы (1) упорядочены по значениям расстояния от корня дерева путей.

Альтернативы формирования дерева кратчайших путей отражаются листьями, путь от корня до которых не обязательно кратчайший, но восстанавливается по правилу построения $p(x, s)$. Обозначим $L(s)$ – множество вершин текущего дерева, L^* – подмножество листьев без постоянной пометки [1]. Очевидно, что в любой момент построения дерева кратчайших путей его узлы можно отобразить на элементы множества

$$L(s) = \bigcup_{x \in N} p(x, s). \quad (2)$$

Расширение дерева кратчайших путей происходит только из некоторого листа без пометки, а листья из множества $L(s) \setminus L^*$ представляют лишь исторический интерес и становятся пассивными. Так как каждому элементу $x, x \in L(s)$, соответствует $d(x)$ – длина кратчайшего пути до корня s , то парами $(x, d(x))$ активные элементы множества L^* представляют приоритетную очередь [1], а пассивные элементы из $L(s) \setminus L^*$ – постепенно растущее дерево кратчайших путей. Обычно на практике $|L(s)| \ll |N|$, что объясняет привлекательность схемы бутстрэппинга.

Представим переменные состояния процесса построения дерева путей элементами отношения $T(x, d, p)$, где x – номер узла дерева (уникальный элемент), d – расстояние от корня s до узла x , p – номер предшествующего узла кратчайшего пути в узел x . По определению, $L(s)$ – проекция отношения $T(x, d, p)$ по атрибуту x .

Очевидно, что состояние процесса построения дерева на любом этапе k представлено тройкой $(x_k, d(x_k), p(x_k))$. Начальное состояние процесса построения дерева с корнем s соответствует тройке $(s, 0, s)$. Условие завершения процесса – $(L^k = \emptyset)$ или $(x_k = t)$, если t – конечная вершина пути. В любом случае $k \leq |L(s)|$ после однонаправленного поиска пути $s \rightarrow t$. Нумерация состояний в явном виде не потребуется, если в операции выборки очередного элемента x_k из L^* контролировать условие завершения процесса. Результат операции $L^k \ominus x_k$ – истинность условия $(L^k \neq \emptyset)$, новое значение x_k в случае $(L^k \neq \emptyset)$ и $k \leftarrow k + 1$. Так как по определению $d(x_k) \leq d(x_{k+1})$, то операция $'\ominus'$ требует упорядочения элементов $L(s)$ по ключу $d(x) \odot x$ (здесь и далее символ $'\odot'$ обозначает операцию конкатенации). Операцию расширения дерева кратчайших путей обозначим символом $'\oplus'$. Результат

операции $L^k \oplus x_k$ – истина, если $x_k \notin L(s)$ и выполнено $L^k \leftarrow L^{k-1} \cup \{x_k\}$. Ее реализация требует упорядочения элементов $L(s)$ по ключу x . Алгоритм Декстры поиска кратчайшего пути $s \rightarrow t$ в терминах определенных выше операций имеет вид:

```
function sp1(s, t) begin
  T1(x, d, p) = (s, 0, s);
  while (L* ⊖ x) ∧ (x ≠ t) do
    foreach (y ∈ x') do
      r = d(x) + w(x, y);
      if (L* ⊕ y) ∨ (d(y) > r) then
        | d(y) = r; p(y) = x
      end
    end
  end
end
```

Здесь предполагается обычно делегируемое системам управления базами данных автоматическое переключение между виртуальными представлениями отношения $T(x, d, p)$ по значениям ключей упорядочения их кортежей.

III. МОДЕЛЬ И АЛГОРИТМ ВСТРЕЧНОГО ПОИСКА

Встречный поиск предлагается проводить на виртуальных представлениях графа, а отношение $T(x, d, p)$ дополнить атрибутами:

$q(x)$ – признак вершины исходного ($q = 1$) или инвертированного ($q = 2$) графа, $x \in L$;

$z(x)$ – признак постоянной пометки вершины графа, $z(x) = (x \in L \setminus L^*)$.

Ключами виртуальных представлений пространства поиска $T(x, d, p, q, z)$ будут $x \odot q(x)$ и $d(x) \odot x \odot q(x)$. Используя сокращения выражений $x \odot q(x) \equiv x_q$, $(3 - q(x)) \equiv \bar{q}(x)$ и $x \odot \bar{q}(x) \equiv \bar{x}_q$, представим алгоритм встречного поиска кратчайшего пути $s \rightarrow t$ в виде:

```
function sp2(s, t) begin
  T1(x, d, p, q, z) = (s, 0, s, 1, false);
  T2(x, d, p, q, z) = (t, 0, t, 2, false);
  while L* ⊖ xq do
    z(xq) = true;
    if (q(xq) = q̄(xq)) ∧ z(xq) then
      | spt(xq);
      | break;
    end
    foreach (y ∈ x'q) do
      r = d(xq) + w(xq, y);
      if L* ⊕ y then
        | d(y) = r; p(y) = xq;
        | q(y) = q(xq); z(y) = false;
      else
        | if (d(y) > r) then
          | | d(y) = r; p(y) = xq
          end
        end
      end
    end
  end
end
```

Здесь источник элементов множества x'_q – исходный или инвертированный граф, указывается признаком $q(x_q)$ вершины ветвления x_q . На уровне хранения такие графы заданы списком дуг $G(x, y, w)$, где w – вес дуги $x \rightarrow y$. Индексируя отношение G по ключу x или y , получаем представление исходного или инвертированного графа в виде структуры смежности $\{(x, x')\}$ или $\{(y, 'y)\}$ (здесь $'y = \{x | w(x, y) \geq 0\}$, где $x, y \in N$).

После достижения некоторой вершины в сопряженном графе процесс расширения встречных деревьев кратчайших путей завершается. Однако для получения оптимального пути необходим перебор всех альтернатив слияния листьев и узлов встречных деревьев с целью поиска вершины v , для которой сумма длин путей $s \rightarrow v$ и $v \rightarrow t$ имеет минимальное значение f :

```
function spt(xq) begin
  v = xq; f = d(v) + d(v̄);
  while L* ⊖ xq do
    r = d(xq) + d(xq̄);
    if f > r then
      | v = xq; f = r;
    end
  end
  do spl(v);
end
```

Из вершины v остается инвертировать путь $v \rightarrow t$ для представления пути $t \rightarrow s$ в виде (1):

```
function spl(x) begin
  d̄ = d(x0); p = p(x1);
  while p ≠ t do
    d(p) ← d̄ + d(x1) - d(p); d̄ = d(p);
    q(p) = 0; p(p) = x;
    x = p; p = p(x1);
  end
end
```

Рассмотренный алгоритм поиска построен на основе компактной реляционной модели, допускающей в реальном времени динамическое изменение структуры и параметров графа. Кардинальность отношений определяется фактическим размером исходных данных и результатов поиска. Очевидно, что расширения модели для учета различных ограничений не требуют изменения предложенной схемы поиска решения.

СПИСОК ЛИТЕРАТУРЫ

1. Ferone, D. Shortest paths on dynamic graphs: a survey/D. Ferone [et al.]/Pesquisa Operacional, 2017. – Vol. 37, iss. 3. – P. 487–508.
2. Ревотюк, М. П. Встречный поиск кратчайших путей на больших динамических графах методом бутстрэппинга/М.П. Ревотюк, Н.В. Хаджинова //BIG DATA and Advanced Analytics: сб. материалов VI Международ. науч.-практ. конф., Минск, 20–21 мая 2020 г. В 2 ч. Ч. 1. – Минск: Бестпринт, 2020. – С. 324–331.

УТОЧНЕНИЕ ДИНАМИЧЕСКОЙ СТРУКТУРЫ ИЗ КОНЕЧНЫХ АВТОМАТОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПОИСКА ШАБЛОНА В ТЕКСТЕ

Савёнок В. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: savionak@gmail.com

В данной работе представлено уточнение динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте. Приведено краткое описание структуры и подробно рассмотрен ее динамический аспект. Описана общая реализация отдельных узлов; представлены схемы переходов конечных автоматов, соответствующих каждому типу узла.

ВВЕДЕНИЕ

Классическим подходом к решению задачи поиска текста по шаблону является использование конечного автомата. Данный подход широко применяется при работе с регулярными и контекстно-свободными грамматиками [1]. Соответствующие им автоматы содержат небольшое число состояний и позволяют сопоставлять достаточно простые шаблоны, в которых применяются выражения вида «последовательность», «вариация» и «повторение». В то же время, выявление шаблонов, зависящих от контекста, требует проектирования автомата, число состояний в котором по отношению к сложности шаблонного выражения растет экспоненциально [2].

Для упрощения процесса разработки таких автоматов в работе [3] предложена декомпозиция на простейшие автоматы, объединенные в многоуровневую динамическую структуру. В текущей работе подробно описан динамический аспект указанной структуры, а также рассмотрена реализация ее отдельных узлов.

ДИНАМИЧЕСКАЯ СТРУКТУРА ИЗ КОНЕЧНЫХ АВТОМАТОВ

Рассмотрим пример динамической структуры автоматов для шаблона $P = ("Microsoft" _ "acquires") \dots [0-3] \dots \{ "Google" _ "Amazon" \}$, изображенный на рисунке 3. В данной структуре автомат, соответствующий отдельному оператору, называется автоматом-оператором. Поиск совпадения литералов также является операцией, для которой создаются автоматы-операторы. Корневому узлу выражения ставится в соответствие автомат-кандидат совпадения шаблона.

Формируемая структура является динамической и изменяется по мере обработки событий. Проследим данный аспект на примере поиска приведенного шаблона в заданной последовательности лексем: «Microsoft», Space, «acquires», Space, «division», Space, «of», Space, «Google», «.».

Представленный шаблон P начинает совпадать на первой лексеме. При этом создается

только два связанных автомата: кандидат и оператор для литерала. После подтверждения совпадения автомат литерала «Microsoft» переходит в заключительное состояние и завершает работу. Осуществляется поиск следующего оператора по дереву выражений, выбирается оператор «последовательность слов» и создается автомат, который привязывается к кандидату. Таким образом, после обработки первой лексемы структура состоит из двух автоматов: кандидата и одного оператора.

При поступлении второй лексемы вызывается текущий автомат-оператор. По лексеме Space принимается решение о продолжении совпадения, но ввиду того, что конструкция не совпала полностью, оператор останется активным.

Лексема «acquires» соответствует литералу в правой части текущего выражения, поэтому для нее создается автомат, который сразу вызывается оператором «последовательность слов». Лексема совпадает. Далее производится поиск следующего оператора по дереву выражений и обнаруживается промежуток в словах. Создается автомат, которому сразу передается событие о совпадении левой части выражения. Поиск в оставшейся последовательности происходит аналогичным образом с постоянной сменой задействованных автоматов.

Таким образом автоматы-операторы верхнего уровня не будут созданы до тех пор, пока не отработают все операторы нижнего уровня. Этот подход снижает накладные расходы на проверку шаблонов, которые позже не смогут продолжить совпадение.

УЗЛЫ ДИНАМИЧЕСКОЙ СТРУКТУРЫ ИЗ КОНЕЧНЫХ АВТОМАТОВ

Автоматы-операторы делятся на два типа: базовые операторы и обработчики отмены совпадения. Диаграмма переходов для базового оператора приведена на рисунке 1.

После создания автомат-оператор находится в состоянии совпадение – Matching. В результате обработки очередного события автомат мо-

жет остаться в том же состоянии либо перейти в одно из заключительных состояний: окончательное совпадение - Final match или отмена совпадения - Rejected.

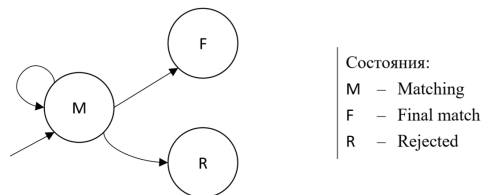


Рис. 1 – Диаграмма переходов базового автомата оператора

Автомат переходит в состояние Final match, если найдено совпадение всего выражения. В этом случае производится поиск вызывающего автомата верхнего уровня по дереву выражений. Если текущий, вызываемый автомат является прямым потомком вызывающего, то последнему передается событие о завершении работы вложенного автомата.

Автомат переходит в состояние Rejected, если не найдено совпадения для следующей лексемы. В такой ситуации генерируется событие отмены совпадения, которое передается ближайшему обработчику в цепочке вызовов автоматов. В динамической структуре обработчиком отмены на самом верхнем уровне выступает кандидат.

Для обработки отмены вводится дополнительное не заключительное состояние неокончательное совпадение – Non-Final match. Автомат переходит в данное состояние в том случае, если обнаружен недостаток данных для принятия решения о совпадении. Таким образом, диаграмма переходов для автомата-обработчика отмены имеет вид, представленный на рисунке 2.

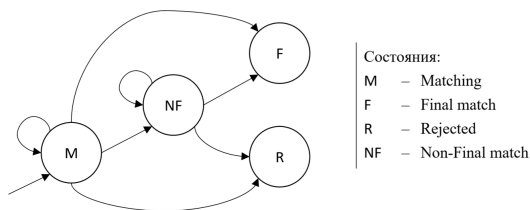


Рис. 2 – Диаграмма переходов автомата обработчика отмены

Автомат переходит в состояние Non-Final match, если вложенные автоматы завершили

проверку совпадения, но хотя бы один из них ожидает прохождения дополнительных проверок в контексте поиска совпадений. К таким операторам относятся контекстные операторы, промежуток в словах и исключения.

При переходе в состояние Non-Final match производятся те же действия, что и при окончательном совпадении, однако, автомат продолжает ожидать события об отмене или окончательном совпадении. При этом он перестает быть вложенным относительно вызывающего его автомата и переходит в подчинение главному автомату системы, что позволяет ему получать ожидаемые события.

Выход из состояния Non-Final match осуществляется либо при прохождении всех дополнительных проверок как вложенными операторами, так и самим обработчиком отмены, либо в случае, когда хотя бы один вложенный автомат или сам обработчик не прошел дополнительные проверки на совпадение.

ЗАКЛЮЧЕНИЕ

Предложен способ реализации элементов динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте. Каждый элемент представлен отдельным конечным автоматом с небольшим числом состояний, что позволяет строить оптимальные по памяти многоуровневые структуры для реализации поиска сложных шаблонов. Формирование и модификация такой структуры происходит по мере совпадения отдельных частей выражения шаблона, что снижает накладные расходы при сопоставлении шаблонов.

1. John E. Hopcroft and Jeffrey D. Ullman Introduction to Automata Theory, Languages, and Computation / John E. Hopcroft [et al.] // Addison-Wesley, 1979. – P. 217.
2. А. С. Морозов Лекции по конечным автоматам и автоматным структурам [Электронный ресурс]. – Режим доступа: <http://math.nsc.ru/~asm256/TA/FANew.pdf>. Дата доступа: 20.03.2020.
3. Савёнок, В. А. Использование динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте / В. А. Савёнок // Компьютерные системы и сети: материалы 56-й научной конференции аспирантов, магистрантов и студентов, Минск, 18 – 20 мая 2020 г. / БГУИР. – Минск, 2020. – С. 95-96.

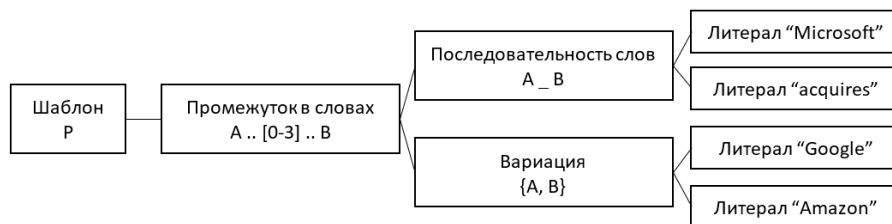


Рис. 3 – Пример структуры из автоматов для дерева выражений

АЛГОРИТМ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

Савич С. И., Гуринович А. Б.

Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: savich.stanislav@mail.ru, gurinovich@bsuir.by

Исследование алгоритмов реализации бизнес-процессами включает в себя как исследование известных решений, так и поиск новых алгоритмов. Актуален анализ методов, оптимизирующих существующие алгоритмы.

ВВЕДЕНИЕ

Основная проблема компаний – это разобщенность подразделений.

В одной компании разные подразделения могут одновременно разрабатывать свои регламенты и автоматизировать свои процессы управления по-разному. В итоге, компании одновременно внедряют разные системы управления, что приводит к формированию среды с разрозненными системами управления, каждая из которых имеет разные подходы, форматы, параметры и требования. Это негативно отражается на эффективности труда сотрудников.

Цель автоматизации процессов – позволить всей компании работать в едином контуре и говорить на едином бизнес-языке.

Автоматизация процессов эффективность работы сотрудников и обеспечивает гибкость и управляемость бизнеса. [1]

В наше время бизнес-процессы являются неотъемлемой частью любого e-commerce продукта. Они несут в себе самую сложную и необходимую функциональность всей системы. Берут на себя обязанности по выполнению и логированию (запись данных о работе программы в специальные файлы логов) всех основных функциональных этапов для бизнеса. Это позволяет существенно оптимизировать процесс выполнения циклических бизнес-процессов при большом объеме данных. [2]

Проблема – отсутствие оптимального алгоритма пакетного распараллеливания задач, что в случае большого количества однотипных входных данных приводит к значительному замедлению работы системы.

I. БИЗНЕС-ПРОЦЕССЫ

Бизнес-процесс – это процесс обработки документа, для которого задана одна точка входа и несколько точек выхода и последовательность действий

(шагов, этапов, функций), совершаемых в заданном порядке и в определенных условиях. [3]

Бизнес-процессы могут быть однократными и циклическими, повторяющимися. Также процессы могут быть укрупненными и дробными. Дроб-

ные процессы могут входить в состав разовых проектов или разных процессов. [4]

При автоматизировании бизнес-процессов необходимо учитывать следующие важные моменты:

- Бизнес-процесс должен быть повторяющимся (циклическим) и конечным, то есть направленным на достижение результата. Отказ – это тоже результат. Моделировать бизнес-процесс для единичного использования всегда нецелесообразно.
- Каждый процесс должен иметь обязательный набор атрибутов: начальный и конечный этапы, человеческие и автоматические компоненты, сроки. [5]

II. АЛГОРИТМ ВЫПОЛНЕНИЯ БИЗНЕС-ПРОЦЕССОВ

Часто встречается ситуация, когда бизнес-процесс достаточно сложный, а запускается он циклично для каждого элемента из входящего списка данных. Таким образом если перед каким-то бизнес-процессом стоит задача обработать каждый элемент из списка размеров в 200 000 элементов, то бизнес-процесс будет вызываться циклично для каждого из них. В качестве примера таких операций в e-commerce платформах можно привести:

1. Обновление цен у продуктов;
2. Формирование статистических данных о продуктах;
3. Генерация файлов фидов (файл, который используется маркетплейсами для получения информации о продаваемых товарах для конкретного магазина);
4. и тд.

Опираясь на описание проблемы выше, можно сделать вывод, что при прямом выполнении алгоритмов, система зависнет на выполнении подобных задач в лучшем случае на час, а в сложных случаях (таких как генерация фидов), можно ожидать задержки в пять-восемь часов. Данная затрата времени для реального бизнеса является недопустимой.

В качестве решения проблемы эффективен алгоритм, оборачивающий бизнес «тяжелые» бизнес-процессы и распределять входные данные на пакеты, которые в свою очередь бу-

дут выполняться в параллельных потоках. Обернуть бизнес процессы можно вручную (непосредственно перед запуском), либо при

помощи аспектного программирования, что скроет реализацию распараллеливания и обеспечит инкапсуляцию. Такой подход позволит сократить время, затрачиваемое на выполнение циклических бизнес-процессов, в несколько раз.

III. ОПИСАНИЕ АЛГОРИТМА

Размер одного пакета данных (количество элементов, которые будут обрабатываться в рамках одной транзакции), количество одновременно выполняемых поток и размер очереди, всегда можно переопределить в зависимости от необходимой задачи или нагрузочных возможностей сервера, на котором будет применяться алгоритм, поэтому все эти данные должны находиться в конфигурациях, что позволит адаптировать алгоритм под нужды конкретной системы и её задач. В качестве значений по умолчанию заданы:

1. Размер пакета – 100 элементов;
2. Количество одновременно выполняемых поток (размер пула) – 100;
3. Размер очереди – 100.

Такой набор значений по умолчанию позволит выполнять бизнес-процессы быстро и также быстро получать первые результаты. При слишком большом размере пула потоков, пользователь системы рискует перегрузить сервер, а при слишком маленьком – производительность вырастет недостаточно сильно, аналогично и для очереди. В случае слишком большого или маленького количества элементов в пакете алгоритм не принесёт значительной выгоды.

ЗАКЛЮЧЕНИЕ

Алгоритм пакетного распараллеливания бизнес-процессов запускается в изолированном потоке (стартовый поток), в котором происходит разбиение входящих данных на пакеты. После формирования пакетов элементов, алгоритм распараллеливает их и начинает выполнение бизнес-процессов в многопоточном (параллельном) режиме. При запуске бизнес-процесса формируется объект, содержащий информацию о выполнении процесса в потоке. Это позволяет начать работать с частью результатов до того, как будут получены абсолютно все. Также стартовым потоком пополняется список результатов до того момента поступления последнего пакета данных. В случае занятости пула и очереди, стартовый поток будет «засыпать» на 100 миллисекунд, а после «пробуждения» будет проверять их на наличие мест для новых задач. Этот алгоритм позволяет избежать переполнения очереди, обеспечивает отказоустойчивость и эффективность системы.

СПИСОК ЛИТЕРАТУРЫ

1. Александров, Д. В. Методы и модели информационного менеджмента / Д. В. Александров, А. В. Костров, Р. И. Макаров, Е. Р. Хорошева; под ред. А. В. Кострова. – М. : Финансы и статистика, 2007. – 336 с.
2. Холстед, М. Х. Начала науки о программах / М.: Финансы и статистика, 1981.
3. Елиферов, В. Г. Бизнес-процессы: регламентация и управление. / В. Г. Елиферов, В. В. Репин. – М.: ИНФРА-М, 2005. – 319 с.
4. Марка, Д. Методология структурного анализа и проектирования. / Давид Марка, Клемент МакГоуэн – М. : ИНФРА-М, 1993. – 240 с.
5. Миронов, А. М. Теория процессов / А. М. Миронов – Переславль-Залесский: Университет г. Переславля, 2008.

РАЗРАБОТКА ТЕСТИРУЮЩЕЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ИЗОЛЯЦИИ ПРОЦЕССОВ

Удовин И. А., Воронова В. В.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: wilcot@ya.ru, veronika.voronova31@gmail.com

В данной статье рассматривается разработка тестирующей системы с использованием современных технологий изоляции процессов. Проанализирована предметная область применения программного комплекса, выделены функциональные требования. Описаны проблемы и их возможные решения, связанные с изолированным запуском кода на операционной системе Linux.

ВВЕДЕНИЕ

С развитием информационных технологий, которые требуют глубоких знаний для их применения, появляется все большая потребность в высококвалифицированных специалистах в сфере ИТ.

Как правило, знакомство с информационными технологиями начинается уже со школы. Кроме теоретической составляющей, значительную роль играет практическая часть. Школьникам предлагают писать программы, запускать их на компьютере и наблюдать за полученными результатами. С этого момента важно заложить понимание, что с помощью программ можно и нужно решать большее количество практических задач.

Олимпиады по спортивному программированию, в большинстве случаев, являются практическими и требуют написания кода, решающего поставленные задачи. При проверке решений возникает потребность в автоматизированных тестирующих системах. Хотя корректность программы можно проверить и вручную, использование автоматического способа крайне необходимо для избежания ошибок, возникающих в следствии человеческого фактора, а так же для ускорения процесса. Олимпиады по информатике как раз позволяют проверять решения в автоматическом режиме, ведь программы участников можно компилировать и запускать без непосредственного участия учителя.

В тоже время, тестирующие системы могут применяться не только для проведения школьных олимпиад, но и для проверки лабораторных работ в университетах, при собеседованиях в компании или при проведении соревнований различного рода.

I. ОПИСАНИЕ ТРЕБОВАНИЙ СИСТЕМЫ

Основной задачей тестирующей системы можно определить возможность проведения соревнований и собеседований. Можно выделить две основные группы пользователей системы:

участники и организаторы соревнований, а также администратор.

Минимально-требуемыми функциями с точки зрения участника либо организатора любой тестирующей системы можно выделить следующие:

- регистрация и авторизация;
- создание и редактирование соревнований;
- создание и редактирование пакетов задач;
- разделение пользователей по правам и ролям, управление доступом пользователей;
- просмотр условий задач;
- отправка решений по задачам;
- получение результатов тестирования решения.

Администратор в свою очередь может управлять списком доступных языков программирования, а также настраивать компиляцию и запуск решений, написанных на языке, которого еще нет в системе.

II. ОСНОВНЫЕ КОМПОНЕНТЫ СИСТЕМЫ

Для того, чтобы система могла безотказно выполнять необходимые функции, а также имела возможность горизонтально масштабироваться, она может быть разделена на следующие два компонента:

- web-интерфейс тестирующей системы – интерфейс, с которым взаимодействует пользователь;
- демон асинхронных операций (далее Invoker) – программа, предназначенная для тестирования посылок пользователей в изолированной среде.

Для работы тестирующей системы необходима база данных для хранения следующей информации:

- информация о пользователях, такая как логины, хеши паролей;
- информация о соревнованиях (название, тип, описание, список задач);
- информация по посылкам пользователей (вердикт, использованное время и память);
- текущее состояние очереди посылок.

К примеру, может использоваться PostgreSQL - объектно-реляционная система управления базами данных [1].

Для работы Invokeг-а необходима файловая система для хранения следующей информации:

- пакетов задач, состоящих из ручных тестов, чекеров, валидаторов и генераторов;
- результатов тестирования каждой посылки: вывод на этапе компиляции, выходные данные программы.

В этих целях удобно использовать распределенную файловую систему CephFS.

III. ПАКЕТ ЗАДАЧИ И ПРОВЕРКА РЕШЕНИЯ

Пакет задачи представляет из себя архив данных со следующим содержанием:

1. чекер – программа, проверяющая, что выходные данные программы соответствуют выходным данным теста;
2. валидатор – программа, проверяющая, что входные данные тестов соответствуют условию задачи;
3. генератор – программа, генерирующая входные данные для тестов;
4. набор тестов – входные и ожидаемые выходные данные.

В случае, если для генерации теста используется генератор, то вместо входных данных указываются параметры запуска генератора.

Так как решение некоторой задачи представляет из себя исходный код программы, то для проверки его корректности необходимо запустить программу на наборе тестов из пакета задачи. Проверка решения состоит из следующих этапов:

1. компиляция исходного кода генераторов, чекеров и валидаторов;
2. компиляция исходного кода программы решения задачи;
3. генерация тестов;
4. запуск исполняемого файла на наборе тестов;
5. анализ результатов выполнения программы.

Каждый из этих этапов подразумевает запуск исполняемого кода. Во время компиляции это запуск компилятора, во время тестирования – запуск решения, во время анализа результатов – запуск чекера. Выполнение этих программ не должно влиять друг на друга, поэтому возникает необходимость в их изоляции.

IV. ИЗОЛИРОВАННЫЙ ЗАПУСК ПРОГРАММ

Для того, чтобы запускать программы в изолированной среде, можно использовать технологии виртуализации, что позволит безопасно запускать программы. Однако данный подход имеет большие накладные расходы из-за необходимости запуска операционной системы в этой среде. Это сильно замедляет тестирование решений и увеличивает потребление ресурсов.

Начиная с версии 3.8 ядра Linux, операционная система позволяет запускать программы в изолированной среде без использования технологии виртуализации. Это стало возможным благодаря Linux namespaces – пространства имен, в которых запускаются программы [2].

При запуске любой программы, Invokeг создает отдельные пространства имен: IPC, MOUNT, NET, PID, USER и UTS. Для большей безопасности, Invokeг запускается от обычного пользователя (не от суперпользователя root). В таком случае нам становятся недоступны некоторые необходимые нам команды, например mount. Поэтому в начале производится инициализация пространства имен USER. Это позволяет получить нам root-права внутри изолированной среды. После этого можно спокойно инициализировать оставшиеся пространства имен [3].

Для того, чтобы не нужно было для каждого теста заново инициализировать все пространства имен, они инициализируются один раз перед запуском всех тестов. Теперь программа решения задачи помещается в заранее инициализированные пространства имен и запускается на каждом тесте. Такой подход позволяет значительно ускорить тестирование, однако возникает риск, что запуск одного теста может повлиять на запуск других, так как изменения в изолированной файловой системе остаются. Чтобы этого избежать и не создавать заново пространство имен MOUNT для каждого теста, используется виртуальная файловая система OverlayFS. Она позволяет сохранить исходные версии файлов, а все изменения помещает в отдельный каталог, который перед каждым тестом можно отчистить [4].

ЗАКЛЮЧЕНИЕ

Разработанное программное средство позволяет проводить собеседования и олимпиады, где в качестве решения предлагается написание программ. Для компиляции и запуска программ используются современные технологии изоляции процессов, что позволяет избежать последствий запуска вредоносного кода, а также ускорить тестирование программ за счет уменьшения накладных расходов благодаря использованию технологий виртуализации.

СПИСОК ЛИТЕРАТУРЫ

1. Новиков, Б. А. Основы технологий баз данных: учебное пособие / Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева; под ред. Е. В. Рогова. – 2-е издание – М.: ДМК Пресс, 2020. – С. 33–34. – ISBN 978-5-97060-841-8.
2. Namespaces in operation, part 1: namespaces overview [Электронный ресурс] / LWN.net – Режим доступа: <https://lwn.net/Articles/531114/>. – Дата доступа: 26.09.2020.
3. User namespaces progress [Электронный ресурс] / LWN.net – Режим доступа: <https://lwn.net/Articles/528078/>. – Дата доступа: 26.09.2020.
4. Wang, K. C. Systems Programming in Unix/Linux – Springer, 2018. – С. 119–120. – ISBN 978-3-319-92428-1

ДЕРЕВЬЯ РЕШЕНИЙ КАК МЕТОД ДИАГНОСТИКИ ЗЛОКАЧЕСТВЕННЫХ НОВООБРАЗОВАНИЙ

Хачатрян А. Г.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: arman_vir@mail.ru

В данной статье представлены актуальность и обоснование выбора данной тематики. Рассматривается метод классификации объектов под название «Деревья решений», а так же описаны главные преимущества выбранного метода. Предоставлены результаты основных этапов экспериментальной части исследования. И сделано заключение на основе всей проделанной работы.

ВВЕДЕНИЕ

Меланома кожи является одной из самых опасных злокачественных новообразований человека. Показатели заболеваемости увеличиваются в 2 раза среди населения Европы каждые 10-20 лет (ежегодный прирост составляет 3% - 7%) [1]. В Европе меланома занимает 17 место среди наиболее часто диагностируемых видов рака у мужчин и 8 место у женщин [2]. Со временем о меланоме сложилось мнение как о заболевании с вариабельным и неблагоприятным клиническим протеканием. В отличие от других злокачественных новообразований, меланома чаще поражает людей относительно молодого возраста, обладает высоким метастатическим потенциалом и резистентностью к лучевой и химиотерапии. Хотя только 4% от всех злокачественных новообразований кожи является меланома, на неё приходится 79% смертей этой группы. Несмотря на это, локальная меланома не является болезнью с однозначно плохим прогнозом. В случае ранней диагностики, поддающимися лечению оказываются до 95% опухолей.

I. АКТУАЛЬНОСТЬ

Лечение больных меланомой кожи является одной из самых актуальных проблем онкологии. Среди нерешенных вопросов в лечении больных меланомой кожи остаётся и вопрос о необходимости проведения профилактической регионарной хирургической операции по удалению лимфоузлов.

На протяжении долгого времени учёные пытаются определить факторы, которые влияют на протекание и точность прогнозов заболевания у больных меланомой кожи. В 2002 г утвердили новую версию классификации меланомы кожи [3]. Изменения были произведены на основании анализа прогностических факторов, изученных у 17 600 больных меланомой, и коснулись следующих параметров:

- толщина опухоли, наличие или отсутствие изъязвления;

- количество метастатически поражённых лимфоузлов и наличие микро- или макро-метастазов в лимфоузлах;
- наличие сателлитов вокруг первичной опухоли и транзиторных метастазов (было объединено в единую III стадию заболевания);
- дополнительную информацию о стадии заболевания получают при лимфатическом картировании и биопсии сторожевого лимфоузла.

II. ОПИСАНИЕ МЕТОДА

Деревья решений – метод, который позволяет устанавливать принадлежность объектов к тому или иному классу зависимой категориальной переменной, измеренной в списке наименований, в зависимости от сопутствующих значений одной или нескольких предикативных переменных [4]. Главной целью построения дерева классификации является построение решающих правил, позволяющих классифицировать объекты. Способность деревьев классификации выполнять параллельное ветвление для анализа вклада отдельных переменных дает возможность работать с предикативными переменными различных типов – как с категориальными, так и с непрерывными. Деревья классификации могут быть очень сложными. Однако графическое представление, позволяет упростить понимание результатов даже для довольно сложных деревьев. Возможность графического представления результатов и простота интерпретации во многом объясняют достаточно большую распространённость деревьев классификации в медицине, а также и в других областях. Процесс построения дерева классификации делится на 4 этапа:

1. определение критерия точности прогноза;
2. определение момента остановки ветвления;
3. выбор вариантов ветвления;
4. определение «необходимого размера» дерева.

Целью проведения анализа с помощью деревьев классификации, является получение максимально точного прогноза. Наиболее точным прогно-

зом является тот, который с наименьшей долей вероятности неправильно классифицирует наблюдения (цена). Соответственно самый лучший прогноз – это тот, который дает наименьшую цену [4].

III. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

В исследование вошёл 716 больной первичной меланомой кожи с локализацией опухоли на коже туловища и конечностей, получавший лечение в НИИОМР за период 1965-2002 гг. В основу деления больных на группы был положен факт развития метастазов в регионарных лимфоузлах. Средний возраст больных, включенных в исследование, составил 49 лет (в пределах 10-84 года). В обеих группах преобладали женщины. В первой группе больные меланомой с метастазами (19,4%), во второй – без метастазов (80,6%). Для определения влияния факторов на развития метастазов в регионарных лимфатических узлах было отобрано 23 клинико-морфологических признака, такие как пол, возраст, кровоточивость опухоли, толщина опухоли по Breslow и т.д.

Для отнесения пациентов к одной из групп «с метастазами» или «без метастазов», использовались решающие правила, построенные с использованием деревьев решений. Метод классификации деревьев решений позволяет предсказывать принадлежность наблюдений к тому или иному классу категориального признака в зависимости от соответствующих значений одного или нескольких предикторных признаков.

Для построения дерева решений использовался алгоритм C&RT .

В алгоритме C&RT каждая вершина дерева представляет собой правило классификации связанное с одним признаком, который наилучшим образом разделяет обучающую выборку на два класса. Для выбора наилучшего из всех возможных вариантов ветвления (т.е. выбора признаков, по которым строилось дерево решений) использовался критерий Джини . По критерию Джини в качестве признака ветвления выбирается такой признак, который минимизирует значение критерия в данной вершине.

Таблица 1 – Важность признаков для классификации

Название	Ранг	Важность
Длительность заболевания до начала лечения	100	1.000000
Площадь	96	0.958123
Толщина опухоли по Breslow, мм	81	0.805809
Уровень инвазии по Clark	44	0.440096
Изъязвление	32	0.323073
Код роста опухоли	20	0.204083
Врастание опухоли в сосуды	19	0.191236

Выбор оптимального размера дерева решений определялся с помощью кросс-проверки. Этот вид проверки рекомендуется использовать в том случае, когда нет отдельной экзаменационной выборки, а обучающая выборка является недостаточно большой, чтобы из неё можно было выделить экзаменационную . В нашем исследовании применялась 3-кратная кросс-проверка. При такой процедуре из обучающей выборки случайным образом формировались 3 подвыборки такого же объёма, как и обучающая выборка (использовался подход построения выборки с возвращением). Дерево решений нужного размера строилось 3 раза, причем каждый раз по очереди одна из подвыборок не использовалась при его построении, а рассматривалась как экзаменационная и для неё вычислялась доля ошибочной классификации. Доли ошибочной классификации по трём экзаменационным выборкам усреднялись и, таким образом, определялась доля ошибочной классификации 3-кратной кросс-проверки.

Таблица 2 – Результаты эксперимента

Класс(всего)	Правильно	Неправильно
С метастазами(139)	111	28
Без метастазов(577)	439	138

Таким образом из таблицы 2 можно увидеть что метод «Дереья решений» правильно классифицировал 80%(диагностическая чувствительность) больных с метастазами, 76%(диагностическая специфичность) больных без метастазов и 77%(диагностическая эффективность) от общего количества больных.

ЗАКЛЮЧЕНИЕ

В данной статье рассматривается метод «Дереья решений», для классификации больных меланомой кожи на две группы:

- больные меланомой с метастазами;
- больные меланомой без метастазов.

Показана важность выбора информативных признаков и размера дерева при построении решающего правила для классификации.

СПИСОК ЛИТЕРАТУРЫ

1. Osterlind A. Epidemiology on malignant melanoma in Europe. // Acta. Oncol. – 1992. – Vol. 31, № 8. – P. 903–908.
2. CLOBOCAN 2000: cancer incidence, mortality and prevalence worldwide, version 1.0. Ferlay J., Bray F., Pisani P., Parkin DM. // IARC cancerbase – 2001. – №.5.
3. Prognostic factors analysis of 17,600 melanoma patients. Validation of the AJCC melanoma staging system. Balch C.M., Soong S.J., Gershenwald J.E., et al. // J. Clin. Oncol. – 2001. – Vol. 19. – P. 3622-3634.
4. Деревья классификации как метод определения тяжести больных, страдающих хронической сердечной недостаточностью. А.А. Халафян, Л.Н. Елисева, С.Г. Сафонова, Савенко Д. В. – 2006. – № 11

ВЕРИФИКАЦИЯ ЛОГИЧЕСКИХ КМОП СХЕМ, УСТОЙЧИВЫХ К МЯГКИМ ОТКАЗАМ

Черемисинова Л. Д., Черемисинов Д. И.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: {cld, cher}@newman.bas-net.by

Рассматривается задача, возникающая при верификации топологии КМОП заказных СБИС. Верификация топологии включает в себя экстракцию (восстановление) электрической схемы из описания топологии и ее сравнение с описанием, исходным для логического синтеза. Предлагается метод верификации эстагирированных КМОП схем, устойчивых к мягким отказам.

ВВЕДЕНИЕ

Сокращение размеров КМОП-транзисторов СБИС, которое связано с технологическим переходом к нормам размеров порядка десятков нанометров резко снизили надежность СБИС. Причина спада надежности существенно связана с мягкими ошибками (soft errors) [1], вызванными, с одной стороны, альфа-частицами, образующимися из-за радиоактивных следов изотопов, имеющих в материалах корпусов и выводах микросхемы, а с другой стороны, атмосферными нейтронами, созданными взаимодействием космических лучей с атмосферой.

В процессе проектирования СБИС сеть КМОП-транзисторов, устойчивая к мягким отказам, строится в результате последовательного применения операций логического синтеза и отображения логической сети в технологический базис. Важным аспектом процесса проектирования СБИС является проверка того, что сеть КМОП-транзисторов правильно реализует исходную систему логических функций.

В работе рассматривается задача верификации, в которой устанавливается сводимость (эквивалентность) исходной для синтеза системы логических функций и сети КМОП-транзисторов, устойчивой к мягким ошибкам, заданной в формате SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами.

I. МЯГКИЕ ОШИБКИ

Когда элемент полупроводниковой структуры СБИС, обычно сток отключенного КМОП-транзистора, находится вблизи дорожки ионизации электрически заряженной частицы, он собирает значительную часть носителей (дырок или электронов), что приводит к импульсу тока. Эффект этого импульса зависит от типа ячейки, к которой принадлежит пораженный элемент. В ячейке памяти или триггере достаточно сильный импульс изменит состояние ячейки, что приведет к одиночному сбою (Single-event Upset, SEU). Элемент памяти при этом остается работоспособным. Одиночный сбой является наиболее распространенным и наименее опасным последствием

ионизирующего воздействия и наиболее типичным случаем мягких ошибок. Когда ионизирующее излучение попадает в область логического вентиля, импульс переходного тока преобразуется в импульс напряжения на выходе – происходит случайный сбой переключения (Single Event Transient, SET), последствия которого зависят от импульса тока и от характеристик нагрузки вентиля. Затем, если длительность импульса напряжения больше времени срабатывания последующих вентилях, он может распространиться по одному или нескольким путям комбинационной логики и достичь триггеров.

Исследования последнего времени показали, что по мере того, как в КМОП технологии продолжают уменьшаться нормы размеров, частота возникновения мягких ошибок комбинационной логики будет расти, станет сопоставимой и, в конечном итоге, более серьезной, чем ошибки элементов памяти [1].

II. УСТОЙЧИВОСТЬ ЛОГИЧЕСКИХ КМОП СХЕМ К МЯГКИМ ОШИБКАМ

Наиболее распространенным стилем логики КМОП схем является логическая комплементарная МОП-структура. Комплементарные МОП-структуры относятся к широкому классу логических схем, называемых статическими схемами, в которых в любой момент времени выход каждого элемента соединяется либо с шиной питания, либо с шиной земли через тракт с малым сопротивлением.

Статический КМОП-вентиль состоит из двух блоков, которые разделены выходным полюсом. Блок, содержащий NMOS транзисторы (pull-down network), размещен между выходом и цепью нулевого потенциала. Блок, содержащий PMOS транзисторы (pull-up network), размещен между цепью напряжения питания и выходом. При этом затворы всех МОП транзисторов связаны с входами схемы. Логическая функция КМОП-вентиля определяется отрицанием функции проводимости МОП транзисторов NMOS блока.

Способность статического КМОП-вентиля противостоять случайным сбоям переключения

зависит от инжектируемого ионизирующими частицами заряда и размеров транзисторов. Увеличение размеров транзисторов позволяет поглотить на выходе вентиля импульс переходного процесса, вызываемый введенным зарядом, и предотвратить его распространение к следующему логическому вентилю. Таким образом, увеличение размера транзистора является возможным решением для уменьшения восприимчивости схемы к ошибкам логических элементов. Однако этот подход увеличивает задержку и площадь схемы. В схемах с технологическими нормами порядка десятков нанометров этот способ трудно осуществим, поэтому для таких схем повышение отказоустойчивости логических элементов осуществляется использованием специальных стилей реализации логики.

Одним из самых распространенных методов повышения отказоустойчивости является стиль реализации логики, называемый «дифференциальная логика на каскадном усилителе, управляющем переключением напряжения» (Differential Cascode Voltage Switch Logic – DCVSL). Вентиль DCVSL является вариантом статического вентиля, в котором логические вычисления выполняются парой дифференциальных каналов в pull-down схеме из NMOS транзисторов, реализующих комплементарные функции и двух PMOS транзисторов с перекрестными связями в pull-up схеме. Дифференциальные выходы вентиля DCVSL генерируются по дифференциальным входным сигналам. При этом блок NMOS транзисторов pull-down network состоит из двух массивов транзисторов, реализующих взаимно инверсные функции. В вентиле DCVSL всегда активен один из массивов NMOS транзисторов.

Вентиль DCVSL имеет повышенную помехоустойчивость к мягким ошибкам, поскольку только транзисторы PMOS подвержены случайным сбоям переключения (SET) [2].

По сравнению с традиционными методами повышения отказоустойчивости КМОП схем со статическим стилем реализации логики (пространственного или временного дублирования) стиль логики DCVSL обеспечивает гораздо меньшие накладные расходы по энергопотреблению и площади. Преимуществами DCVSL являются низкая нагрузочная емкость на входах, отсутствие статического энергопотребления. Недостатки состоят в необходимости обеспечения дополнительных входов (для инверсий входных сигналов), введения большего числа транзисторов для реализации одной функции.

III. ВЕРИФИКАЦИЯ КМОП СХЕМЫ, УСТОЙЧИВОЙ К МЯГКИМ ОТКАЗАМ

Проверка проектирования на этапе логического синтеза состоит в установлении эквивалентности исходной системы логических функ-

ций и функций, реализуемых сетью КМОП-транзисторов. Представления проектируемого устройства до синтеза и после его выполнения имеют большое семантическое различие моделей. Поэтому невозможно непосредственно проверить их эквивалентность, и нужно преобразовать плоскую сеть КМОП-транзисторов в иерархическую сеть из подсхем, реализующих логические элементы. Эта сеть логических элементов может быть проверена на эквивалентность исходной системе булевых функций. Процесс, в результате которого из плоской схемы строится иерархическая называется декомпиляцией КМОП схемы [3]. Целью декомпиляции является замена представления схемы на низком (транзисторном) уровне схемой на уровне логических элементов. Разработанный декомпилятор использует формат SPICE для представления схем.

Разработана компьютерная программа для автоматического извлечения иерархии из списка соединений цифровой КМОП схемы транзисторного уровня. Программа выполняет следующую последовательность шагов: 1) анализ исходного описания SPICE и построение хеш-таблицы для хранения двудольного помеченного графа схемы; 2) факторизация двудольного помеченного графа на сети транзисторов, связанных по постоянному току (ССС); 3) распознавание СССР, представляющих собой статические и DCVSL КМОП вентиля; 4) нахождение для них соответствующих логических функций в виде логических формул и построение хеш-таблицы для хранения экземпляров КМОП вентиля; 5) построение логической сети из распознанных вентиля и 6) генерация логических уравнений на языке SPICE, реализуемых этой сетью.

Верификация логических уравнений, реализуемых этой сетью, выполняется проверкой их эквивалентности системе логических функций, заданной для синтеза, посредством сведения к задаче проверки выполнимости КНФ. Для этого программным комплексом [3] строится КНФ, выполнимость которой свидетельствует о нарушении реализуемости (эквивалентности) исходного описания порожденным.

IV. СПИСОК ЛИТЕРАТУРЫ

1. Johnston, A. H. Radiation effects in advanced microelectronics technologies / A. H. Johnston // IEEE Trans. Nuclear Science. – 1998. – Vol. 45. – No. 3. – P. 1339–1354.
2. Zhou, Q. Cost-Effective Radiation Hardening Technique for Combinational Logic / Q. Zhou, K. Mohanram // Proc. Int'l Conf. on Computer-Aided Design, 2004. – P. 100–106.
3. Черемисинов, Д.И. Извлечение сети логических элементов из КМОП схемы транзисторного уровня / Д.И. Черемисинов, Л.Д. Черемисинова // Микроэлектроника. – 2019. – Том 48. – № 3. – С. 224–234.

АЛГОРИТМ СОВМЕСТНОГО ОБНАРУЖЕНИЯ - ИЗМЕРЕНИЯ ПАРАМЕТРОВ ПРОТЯЖЕННОЙ ПО ДАЛЬНОСТИ КОРРЕЛИРОВАННОЙ ОШП В РАДИОЛОКАТОРАХ СОПРОВОЖДЕНИЯ

Чигирь И. В.

Учреждение образования «Военная академия Республики Беларусь»

Минск, Республика Беларусь

E-mail: ivan.chigir.a@gmail.com

Для импульсно-доплеровских радиолокаторов сопровождения предложен алгоритм обнаружения протяженной по дальности коррелированной ответной шумовой помехи с оценкой ее радиальной протяженности и ширины спектра частот.

ВВЕДЕНИЕ

Наибольшей эффективностью подавления импульсно-доплеровских радиолокационных станций (ИД РЛС) точного измерения координат обладают комбинированные имитирующие и маскирующие помехи, которая представляет собой аддитивную смесь узкополосной протяженной по дальности ответной шумовой и уводящих помех (УП) по дальности и скорости [1].

Для работы устройства компенсации маскирующей составляющей КИМ-помехи в канале сопровождения цели по дальности, а также более точного сопровождения постановщика такого рода помех по угловым координатам помимо обнаружения ответной шумовой помехи (ОШП) возникает необходимость оценки вектора ее априорно неизвестных параметров $\alpha_h = \|\Delta f_h; T_{0h}\|^T$ [1]. Таким образом, возникают две задачи – обнаружения протяженной по дальности коррелированной ОШП и оценки вектора ее параметров.

I. ПОСТАНОВКА ЗАДАЧИ

В начальный момент постановки КИМ-помехи (t_0) и в течение первой секунды увода, время запаздывания и частота Доплера УП соответствуют времени запаздывания и частоте Доплера ОС $t_r^{up} = t_r^s, F_d^{up} = F_d^s$ [1].

Учитывая сказанное выше, обнаружение КИМ-помехи сводится к обнаружению протяженной по дальности коррелированной ОШП помимо сопровождаемого сигнала (УП и/или ОС), который является мешающим и подлежит декорреляции [1].

II. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

По результатам обработки принятого сигнала формируется матрица $M_{i,j}$, составленная из достаточных статистик в элементах разрешения по времени запаздывания и частоте Доплера (рисунок 1). Количество элементов разрешения по дальности J определяется периодом повторения импульсов T_p и временем дискретизации Δt , количество элементов разрешения по частоте Доплера I определяется количеством когерентно

накапливаемых импульсов, что для случая ИД РЛС соответствует числу импульсов в пачке L .

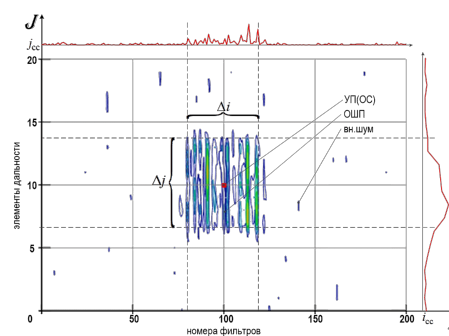


Рис. 1 – Матрица принятия решения

Следящие системы по времени запаздывания и радиальной скорости обеспечивают работу алгоритма обнаружения-измерения ОШП данными о сопровождаемом сигнале (УП и/или ОС) – соответствующие номера элементов дальности и частоты. Далее алгоритм оценивания распараллеливается на два алгоритма – алгоритма оценивания радиальной протяженности ОШП по развертке частотных фильтров в элементе дальности, соответствующего сопровождаемому сигналу (ССг) и алгоритма оценивания ширины полосы частот ОШП по развертке дальности в фильтре, соответствующем ССг (на рисунке 1 данные развертки выделены красным цветом и вынесены за пределы осей ординат и абсцисс). Обнаружение ОШП осуществляется путем сравнения с порогом результатов свертки принятой реализации с прямоугольными эталонами различной длины. Оценка мощности фона σ_{fona}^2 производится в четырех окнах, включающих в себя по Δ_j элементов дальности и Δ_i элементов частоты, с выбором минимума для уменьшения влияния остатков компенсации мешающих отражений и других воздушных целей.

В начале алгоритма открывается цикл по длине прямоугольного эталона $\Delta j(\Delta i)$. Максимально возможная длина эталона ограничена величиной $\Delta j_{max}(\Delta i_{max})$. Величина $\Delta j_{max}(\Delta i_{max})$ выбирается исходя из [1]. После открытия цикла по $\Delta j(\Delta i)$ вычисляется адаптивный порог об-

наружения $Z_{i(j)*} = \sigma_{fona}^2 \cdot \lambda_{i(j)} \cdot \sqrt{\Delta j(\Delta i)}$. Величина относительного порога $\lambda_{i(j)}$ вычисляется согласно [2] с учетом размерности $\Delta j(\Delta i)$. Затем открывается цикл по дискретам окна частоты (дальности) $I(J)$. В этом цикле производится скользящее суммирование $i(j)$ дискретов реализации и сравнение их с порогом $Z_{i(j)}$. С порогом сравниваются лишь значения скользящих сумм при $\Delta j(\Delta i) \geq \Delta j_{min}(\Delta i_{min})$. Величина $\Delta j_{min}(\Delta i_{min})$ выбирается исходя из [1]. Местоположение (номер начального дискрета $i(j)$) прямоугольника, его длина и величина $\Delta j(\Delta i)$ скользящей суммы запоминаются, если последняя превышает порог. По окончании цикла по $i(j)$, скользящие суммы (амплитуды), которые были записаны в оперативно-запоминающее устройство (ОЗУ) в течении предыдущих циклов по $\Delta j(\Delta i)$, пересчитываются для сопоставления со вновь обнаруженными целями (участками целей) при следующем значении длины прямоугольника $\Delta j(\Delta i)$. На выходе алгоритмов формируются оценки начальных элементов частоты (дальности) – $j_{min}(\hat{i}_{min})$, соответствующих ОШП и протяженности по частоте и дальности – $\hat{\Delta j}(\hat{\Delta j})$.

Далее происходит формирование результирующей статистики для обеспечения работы алгоритма обнаружения. Формирование решающей статистики производится по выражению

$$Z = \sum_{k=i_{min}}^{i_{min}+\Delta i} \sum_{l=j_{min}}^{j_{min}+\Delta j} M_{k,l}, \quad (1)$$

которое сравнивается с порогом обнаружения $Z_* = \hat{\Delta i} \cdot \hat{\Delta j} \cdot \lambda_{i,j}(F, \hat{\Delta i}, \hat{\Delta j})$. Величина относительного порога $\lambda_{i,j}(F, \hat{\Delta i}, \hat{\Delta j})$ вычислялась согласно [2, стр.239].

Для анализа показателей качества синтезированного устройства обнаружения-измерения ОШП было проведено математическое моделирование. Принятый сигнал представлял собой аддитивную смесь протяженной по дальности коррелированной ОШП и внутреннего шума, уводящая помеха отсутствует для того чтобы рассчитать предельные характеристики обнаружения. Вектор параметров ОШП считается известным, поэтому апостериорная среднеквадратическая ошибка равна нулю. В качестве критерия оптимальности системы обнаружения был выбран критерий Неймана-Пирсона. На выходе данной системы оценивались характеристики обнаружения, представляющие собой зависимость условной вероятности правильного обнаружения от отношения ОШП-шум при заданной условной вероятности ложной тревоги.

Моделирование проводилось для трех значений полосы спектра ОШП – 7, 10 и 20 кГц. Отношение мощности ОШП к мощности внутренних шумов задавалось на выходе устройства внутрипериодной обработки и изменялось в диапазоне от 0,01 до 1,0 с шагом 0,001. Количество модельных экспериментов 10^5 , что позволило обеспечить доверительную вероятность 0,9 и относительную погрешность 10 %.

По результатам обработки формировалась матрица принятия решений, сформированная из выходных сигналов 5, 10 и 14 взаимно расстроенных фильтров (в зависимости от Δf_h) с учетом квадратичного детектирования и некогерентного суммирования в 20 внутрипериодных каналах дальности. Линейка фильтров перекрывала однозначный диапазон доплеровских частот от 0 до 100 кГц при полосе одного фильтра примерно 7, 10 и 20 кГц соответственно. Характеристики обнаружения оценивались в известном элементе дальности и номере фильтра, где подсчитывались результаты превышения порога. Оценка уровня фона производилась в 4 отсчетах дальности в 9 различных фильтрах. Расчетные характеристики обнаружения (по результатам моделирования) представлены на рисунке 2 в виде штрих-пунктирных линий для вероятности ложной тревоги равной 0,001.

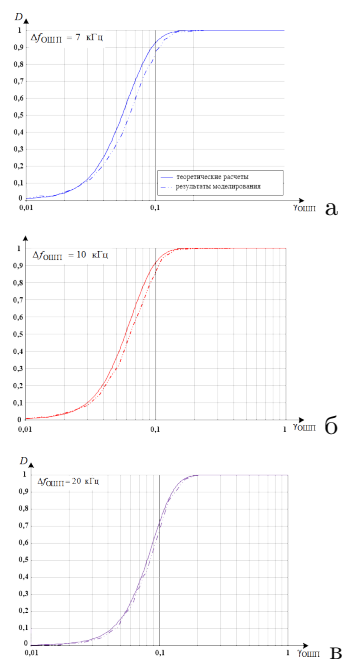


Рис. 2 – Характеристики обнаружения

ЗАКЛЮЧЕНИЕ

По результатам моделирования установлена адекватность математической модели. Точность оценивания α_h снижается при отношениях ОШП-шум менее 5-15 дБ.

СПИСОК ЛИТЕРАТУРЫ

1. Чигирь, И. В. Анализ воздействия комбинированной имитирующей и маскирующей помехи на системы сопровождения по дальности и скорости радиолокаторов точного измерения координат / И. В. Чигирь, С. А. Горшков, Н. К. Кузьмичев // Вестн. ВАРБ. – 2018. – № 1 (58). – С. 71–81.
2. Охрименко, А. Е. Основы радиолокации и радиоэлектронная борьба / А. Е. Охрименко. – Ч. 1. Основы радиолокации. – М. : Воениздат, 1983. – 457 с.

СТРУКТУРЫ ПРОСТРАНСТВЕННЫХ ДАННЫХ В ГЕОИНФОРМАЦИОННЫХ СИСТЕМАХ

Шурмилов П. К.

Кафедра систем управления, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: pkeycalculator@gmail.com

В данной статье автором рассматриваются основные структуры пространственных данных, характеризующих пространственные объекты в геоинформационных системах.

ВВЕДЕНИЕ

Объектом информационного моделирования в геоинформационных системах является пространственный объект. Некоторое множество цифровых данных о пространственных объектах образует пространственные данные. В практике геоинформатики определен набор базовых моделей пространственных данных: растровая, регулярно-ячеистая, квадродерево, векторная [1].

I. ХРАНЕНИЕ РАСТРОВЫХ ДАННЫХ

Растры применяются для хранения и обработки данных дистанционного зондирования, для представления цифровых моделей рельефа, при визуализации геоданлогоритмы. Существует множество вариантов кодирования растровых структур. Некоторые из них более экономно расходуют память, другие позволяют получать более быстрые алгоритмы. Растровая модель соответствует двумерному ячеистому изображению, которое хранится в памяти компьютера в виде одномерной последовательности значений. Растровые изображения обычно разлагаются по строке сверху – слева. Далее будут описаны другие способы эффективного представления растров [2]. В некоторых форматах графических файлов используется сжатие изображения, основанное на замене длительных последовательностей повторяющихся значений парой <значение, количество повторов> (см. рис. 1-а). Географические данные обычно автокоррелированы. В растровой модели это означает, что соседние ячейки имеют большую вероятность быть одинаковыми, чем разобценные. При обычном порядке сканирования в конце каждой строки происходит скачок на начало следующей строки. Предложим простое изменение порядка сканирования. Нечетные строки будем кодировать слева направо, а четные – в обратном направлении (см. рис. 1-б). Направление сканирования напоминает движение быка, вспахивающего поле. Отсюда название этого способа сканирования – бустрофедон (греч. – бык, вспахивающий поле). Теперь при переходе к новой строке первая ячейка является смежной последней ячейке старой строки. Так в линейном разложении растра

сохраняется автокорреляция и повышается эффективность кодирования [3].

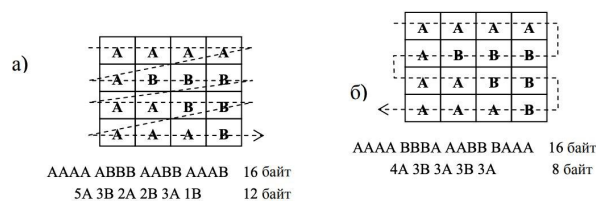


Рис. 1 – Порядки сканирования растров, их линейное разложение и сжатие: а) обычный порядок; б) бустрофедон

Порядок сканирования Мортон основан на иерархическом разбиении карты. В предыдущих способах сканирования учитывалась автокорреляция значений ячеек только по одному направлению (по строке). Географические объекты образуют на растровом изображении пятна. В порядке Мортон предпринимается попытка сканирования ячеек таким образом, чтобы охватить линией обхода эти двумерные пятна.

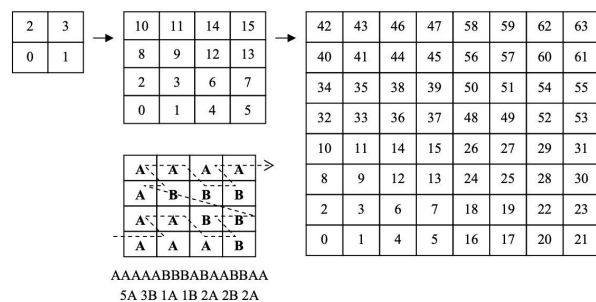


Рис. 2 – Порядок сканирования растра по Мортону

II. ИЕРАРХИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

Рассмотренные выше порядки сканирования растровых изображений дают незначительные различия в компрессии данных. Основное преимущество Мортон-сканирования и других иерархических структур данных заключается в более быстром доступе к данным. Информация распределена по карте неравномерно. Увеличение разрешения растрового изображения приводит к увеличению размеров файлов, а уменьшение – к потере информации. Далее пойдет речь

об адаптивных методах представления растровых данных с разной плотностью информации. На рис. 4 изображена растровая матрица размера 16 x 16, в которой содержатся 255 значений "А" и одно "В". Индексируем растр следующим способом. Разделим матрицу на четыре подматрицы размера 8 x 8 и нумеруем их 0, 1, 2, 3 в порядке Мортона. Назовем подматрицу гомогенной, если в ней содержатся одинаковые значения. Будем рекурсивно разбивать негомогенные подматрицы до тех пор, пока не достигнем гомогенности всех подматриц. Таким способом получим адаптивное разрешение растрового изображения, где участки с меньшей плотностью информации представлены крупными блоками ячеек, а с большей плотностью – мелкими блоками ячеек. Идея выделения гомогенных блоков растра тождественна кодированию растра по Мортону. Гомогенный блок растра размера $m \times m$ при сканировании по Мортону соответствует коду $\langle m_2, A \rangle$

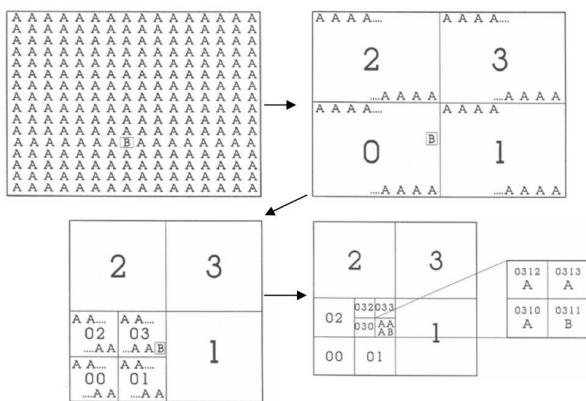


Рис. 3 – Разбиение растра на гомогенные блоки

Соответствие двумерных растровых координат ячейки и адреса ячейки в последовательном файле похоже на аналогичное преобразование при кодировании растра по Мортону. Единственное отличие в том, что используется система счисления с основанием четыре. В примере на рис. 4 ячейка "В" имеет код 0311. В двоичной системе счисления $N = 0311 = (00110101)_2$. Разделим биты между растровыми координатами и выясним, что ячейка лежит в четвертой строке и седьмом столбце. Представленные таким способом растровые данные соответствуют квадродереву, вершина которого – исходное изображение, а листья – гомогенные блоки ячеек. При кодировании квадродеревьев ячейки на каждом уровне могут содержать либо значение гомогенного блока, либо указатель на следующий уровень. Дерево, показанное на рис. 5 может быть представлено в виде линейной последовательности следующим образом.

Позиция:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Содержание:	2	6	А	А	А	А	А	А	10	А	14	А	А	А	В	А	А
Уровень:	0	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4

Рис. 4 – Кодирование квадродерева

Как уже отмечалось выше, основное преимущество иерархической организации данных в ГИС заключается в пространственном упорядочении информации и более быстром ее поиске. Поэтому рассматриваются две задачи ГИС, связанные с индексированием квадродеревьями: первая – поиск всех частей карты с заданным значением и вторая – определение содержимого некоторой ячейки [4].

Обозначим n – число уровней квадродерева (тогда размер растра $2n \times 2n$) и через m – число листьев в дереве. Чтобы найти части карты с некоторым значением B , необходимо проверить каждый лист дерева, что потребует m шагов. Определение значения ячейки происходит путем спуска по квадродереву до тех пор, пока не будет получен гомогенный блок. В худшем случае, когда ячейка находится на самой вершине дерева (как, например, ячейка B на рис. 4), поиск займет n шагов. Сравним теперь трудоемкости обеих задач на квадродереве с трудоемкостями этих задач при различных вариантах сканирования растра.

Таблица 1 – Трудоемкость алгоритмов при различной организации растров

Структура данных	Поиск частей с заданным значением	Определение значения ячейки
Квадродерево	m	n
Обычный порядок	$4n^{2*}$	1^{**}
Бустрофедон	m^{***}	m^{***}
Мортон	m^{***}	m^{***}

Прим. * – проверяется каждая ячейка матрицы; ** – непосредственное вычисление позиции ячейки; *** – число цепочек примерно соответствует числу листьев; **** – проверяется каждая цепочка.

III. ЗАКЛЮЧЕНИЕ

Были рассмотрены и проанализированы некоторые способы кодирования пространственных данных в геоинформационных системах, а именно: обычный порядок, бустрофедон, кодирование Мортон и квадродерево. Оценена трудоемкость задачи поиска всех частей карты с заданным значением и задачи определения содержимого некоторой ячейки при использовании каждого из вышеуказанных способов.

1. Варфоломеев И.В., Савельев А.С. Представление и обработка пространственных данных в ГИС: Методические указания / Варфоломеев И.В., Савельев А. – Красноярск: КГТУ, 2001.
2. Коновалова Н.В., Капралов Е.Г. Введение в ГИС / Коновалова Н.В., Капралов Е. – Петрозаводск: Издво Петрозаводского Госуниверситета, 1995.
3. Кошкарёв А.В., Тикунов В.С. Геоинформатика / Под ред. Д.В. Лисицкого. М. – Картгеоцентр – Геоиздат, 1993.
4. Якубайлик О.Э. Методы и приемы пространственного анализа в геоинформационных системах: Учебное пособие / Якубайлик О. – Красноярск: Издательство КрасГУ, 2001.

АДАПТАЦИЯ К ПРОСТРАНСТВЕННОЙ ОРИЕНТАЦИИ ОБЪЕКТОВ В ЗАДАЧЕ СИГНАЛЬНО-ТРАЕКТОРНОГО РАСПОЗНАВАНИЯ КЛАССОВ

Ярмолик С. Н., Зайко Е. В., Свинарский М. В., Леонович А. С.

Кафедра автоматики, радиолокации и приемо-передающих устройств, Военная академия РБ

Минск, Республика Беларусь

E-mail: zaiko.eugene@mail.ru, mechislav1993@gmail.com

Основным препятствием практической реализации оптимального устройства распознавания является наличие априорной неопределенности, а также необходимость учета пространственной ориентации объекта при формировании сигнально-траекторных классификационных признаков. В статье предложен вариант адаптации классификационных признаков с помощью метода Монте-Карло.

ВВЕДЕНИЕ

Успех противовоздушного боя существенно зависит от качества решения задач целераспределения и целеуказания. Эффективность решения данных задач предполагает наличие информации о классах или типах наблюдаемых объектов. Факт принадлежности наблюдаемого объекта к определенному классу (типу) устанавливается в процессе решения задачи радиолокационного распознавания (РЛР). С учетом того, что задача РЛР решается в условиях сложной воздушной и помеховой обстановки при ограниченности временного ресурса возникает необходимость совершенствования алгоритмов классификации. Для повышения эффективности РЛР используются комбинирование и комплексирование признаков распознавания, их адаптация к условиям наблюдения объектов, использование последовательных процедур принятия решения и т.д. [2]. Для решения задачи РЛР в качестве классификационных признаков широкое распространение получили радиолокационные портреты (РЛП) объектов наблюдения, а также траекторные признаки, характеризующие закономерности их движения [2]. РЛП представляет собой распределение комплексных амплитуд отраженного сигнала по разрешаемым элементам цели. Траекторные признаки являются результатом фильтрации параметров траектории объекта и определяются особенностями пространственного перемещения объекта относительно радиолокатора [3]. Вопрос совместного использования сигнальных и траекторных признаков с адаптацией к условиям наблюдения востребован и актуален при создании перспективных систем РЛР.

I. ОСНОВНАЯ ЧАСТЬ

При синтезе алгоритмов РЛР наиболее широкое распространение получил Байесовский подход, подразумевающий формирование значения среднего (апостериорного) риска [1]. На основании выражения среднего риска формируется решающее правило для принятия решения о принадлежности наблюдаемого объекта к одному из M классов. Стоит отметить, что общий подход

к формированию среднего риска принятия решения известен [1]. При этом стоит отметить проблему априорной неопределенности, которая возникает при отсутствии данных об априорных плотностях вероятностей (ПВ) классификационных признаков. Сложность практической реализации оптимального устройства, а также отсутствие необходимых априорных данных, обуславливает целесообразность перехода к квазиоптимальным алгоритмам. В квазиоптимальных алгоритмах распознавания процедура принятия решения характеризуется особенностями обработки реализации принятого сигнала на основании выражения для отношения правдоподобия. Рассмотрим особенности формирования отношения правдоподобия при совместном использовании сигнальных и траекторных признаков. При совместном формировании сигнальных и траекторных признаков стоит отметить их слабую корреляцию [2], что позволяет представить обобщенное отношение правдоподобия в следующем виде: $\Lambda(f|A_g) = \Lambda_\xi(f|A_g)L_\eta(f|A_g)$, где $\Lambda_\xi(f|A_g)$ - отношение правдоподобия, характеризующее сигнальные признаки; $L_\eta(f|A_g)$ - коэффициент правдоподобия, характеризующий траекторные признаки. Обработываемые РЛП являются функцией ряда параметров, которые оказывают существенное влияние на вид и параметры РЛП. В качестве одного из таких параметров выступает ориентация наблюдаемого объекта в пространстве [3]. Пространственную ориентацию наблюдаемого объекта будем характеризовать совокупностью трех углов: курса, тангажа и крена ($\Theta^{ЛВ} = \|\psi^{ЛВ}\nu^{ЛВ}\gamma^{ЛВ}\|^T$) в системе координат линии визирования «радиолокатор-объект» [3]. Отношение правдоподобия $\Lambda_\xi(f|A_g)$ принятой реализации f , с учетом углов пространственной ориентации наблюдаемого объекта (УПО) $\Theta^{ЛВ}$, будет иметь вид [3]: $\Lambda_\xi(f|A_g) = \int_{\Gamma_{\Theta^{ЛВ}}} p(\Theta^{ЛВ}|A_g)\Lambda(f|\Theta^{ЛВ}, A_g)d(\Theta^{ЛВ})$, где $\Theta^{ЛВ}$ - УПО объекта в системе координат линии визирования; $p(\Theta^{ЛВ}|A_g)$ - априорная ПВ УПО $\Theta^{ЛВ}$, при наблюдении объекта g -го класса; $\Lambda(f|\Theta^{ЛВ}, A_g)$ - отношение правдоподобия, условное по классу объекта наблюдения A_g и значению его УПО

$\Theta^{\text{ЛВ}}$. Следует отметить, что априорная информация о УПО, для текущих условий наблюдения, зачастую отсутствует. Однако стоит отметить, что сформированные текущие оценки, могут уточняться с учетом использования априорной информации. В связи с этим в качестве априорных ПВ УПО наблюдаемого объекта используются их апостериорные значения ($p(\Theta^{\text{ЛВ}}) \approx p(\Theta^{\text{ЛВ}}|A_g), g = \overline{1, M}$) [3]. При этом апостериорный ЗР УПО объекта $p(\Theta^{\text{ЛВ}})$ связан нелинейным функциональным преобразованием $h_{\Theta^{\text{ЛВ}}}(\dots)$ с распределением результатов оценивания координат и параметров движения наблюдаемого объекта $p(\hat{\alpha})$ ($p(\Theta^{\text{ЛВ}}) = p(h_{\Theta^{\text{ЛВ}}}(\hat{\alpha}))$) [3]. При использовании траекторных признаков, функция правдоподобия $p(f|\hat{\alpha}, A_g)$, характеризуется значениями вектора состояния $\hat{\alpha}$. Составляющие вектора состояния (значения координат объекта, скорости и ускорения их изменения и т.д.) отражают особенности движения наблюдаемого объекта, т.е. траекторные признаки. Траекторные признаки могут быть получены путем функционального преобразования $\hat{\eta} = h_{\eta}(\hat{\alpha})$ от значений вектора состояния объекта $\hat{\alpha}$ [2]. При допущении о гауссовско-марковском изменении вектора траекторных признаков ПВ результирующей оценки будет являться максимально правдоподобной оценкой, полученной рекуррентным способом. Таким образом, выражение для функции правдоподобия будет выглядеть следующим образом [2]: $p(f|A_g) \approx L_{\eta}(f|A_g) = \int_{\Gamma_{\eta}} p(\eta|A_g)p(\hat{\eta}|f, A_g)d(\eta), g = \overline{1, M}$, где η - траекторные признаки объекта; $p(\eta|A_g)$ - условная ПВ параметра η , при наблюдении объекта g -го класса; $p(\hat{\eta}|f, A_g)$ - апостериорная ПВ траекторных признаков наблюдаемого объекта. Анализ совместного использования сигнальных и траекторных признаков в алгоритмах РЛР показывает необходимость формирования апостериорного ЗР УПО объекта $p(\Theta^{\text{ЛВ}})$, а также в получении апостериорной ПВ оценки траекторных признаков $p(\hat{\eta}|f, A_g)$. Для аппроксимации апостериорных ПВ оценок предлагается использовать численный метод Монте-Карло [4]. В основе метода Монте-Карло лежит аппроксимация произвольной плотности $p(\alpha)$ совокупностью случайных отсчетов $\alpha^{(s)} (s = \overline{1, N_{\text{МК}}}, N_{\text{МК}} \gg 1)$ с использованием нормированных весов этих отсчетов $w^{(s)}$ [4]. В качестве примера использования

метода Монте-Карло на рисунке 1 представлен вариант аппроксимации ПВ оценки угла курса и радиальной скорости наблюдаемого объекта. На основании данных оценок составляющих вектора состояния $\hat{\alpha}$, а также значений корреляционной матрицы ошибок фильтрации \mathbf{R} формируются весовые коэффициенты $w^{(s)}$ и значения отсчетов $\alpha^{(s)}$. С учетом полученных, с помощью функциональных преобразований $\hat{V}_r^{(s)} = h_{\eta}(\hat{\alpha}^{(s)})$ и $\hat{\psi}^{(s)} = h_{\Theta^{\text{ЛВ}}}(\hat{\alpha}^{(s)})$, значений случайных отсчетов $\hat{\psi}^{(s)}$ и $\hat{V}_r^{(s)}$, и на основании нормированных весов $w^{(s)}$, аппроксимируется ПВ угла курса $p(\hat{\psi}^{\text{ЛВ}})$ (рис.1,а) наблюдаемого объекта, а также апостериорная ПВ траекторных признаков $p(\hat{V}_r|f, A_g)$ (рис.1,б).

II. ЗАКЛЮЧЕНИЕ

Рассмотрен вариант использования численного метода Монте-Карло при аппроксимации ПВ сигнальных и траекторных признаков в интересах решения задачи РЛР. Представленный подход позволяет решать задачу классификации при совместном использовании сигнальных и траекторных признаков. Использование численного метода Монте-Карло при аппроксимации априорной ПВ УПО наблюдаемого объекта позволяет адаптировать РЛР распознаваемых объектов к УПО объекта наблюдения. Совместное использование сигнальных и траекторных признаков с адаптацией к УПО объекта позволяет повысить эффективность решения задачи распознавания при проектировании устройств РЛР в современных и перспективных радиолокаторах.

III. СПИСОК ЛИТЕРАТУРЫ

1. Репин, В. Г. Статистический синтез при априорной неопределенности и адаптации информационных систем / В. Г. Репин, Г. П. Тартаковский. – М.: Сов. радио, 1977. – 432 с.
2. Горшков, С. А. Сочетание сигнальных и траекторных признаков в задачах измерения-распознавания / С. А. Горшков, А. С. Солонар // Прикладная радиоэлектроника. – 2009. – Т. 8, №4. – С. 497–500.
3. Ярмолик, С. Н. Оценивание углов ориентации летательного аппарата в интересах адаптации к условиям наблюдения / С. Н. Ярмолик [и др.]. // Вест. Воен. акад. Респ. Беларусь. – 2017. – №4. – С. 73–82.
4. Соболев, И. М. Численные методы интегрирования Монте-Карло / И. М. Соболев. – М.: Наука, 1973. – 311 с.

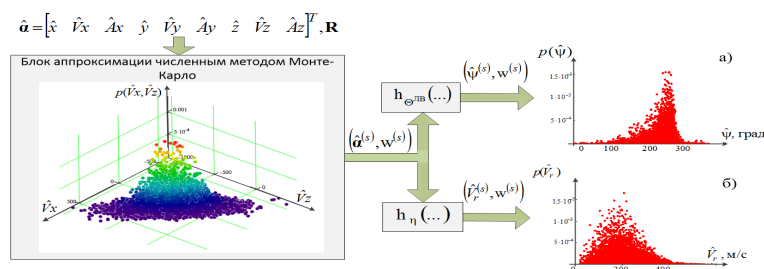


Рис. 1 – Пример аппроксимации плотностей вероятности численным методом Монте-Карло

Содержание

Пленарные доклады	9
Вашкевич М. И., Азаров И. С. «Биоинспирированный и психоакустически мотивированный анализ звуковых сигналов»	10
Таранчук В. Б. «Методы и средства системы ГеоБазаДанных для адаптации компьютерных моделей. Примеры адаптации»	14
Секция «Системы управления»	17
Бейсенби М. А., Бейсембина С. Е. «Управление краткосрочными колебаниями и флуктуацией в развитии основных фондов отрасли экономической системы»	18
Акинфина М. А. «Цифровые технологии управления предприятием»	21
Алина Г. Ж., Марков А. В., Ташатов Н. Н. «Сетевые решения развития систем управления электроэнергетическими системами»	22
Архипенко Я. С., Парамонова А. Е., Белощедов Е. С., Гаруля Д. В. «Сравнительный анализ разновидностей роботов для автоматизации пищевого производства»	24
Волосович С. В. «Использование бессерверных технологий для обработки больших данных»	26
Гринюк Д. А., Олиферович Н. М., Сухорукова И. Г., Оробей И. О. «Настройка ПИД-регулятора через апериодический цифровой регулятор»	28
Дубовик И. А., Бойкачев П. В., Исаев В. О. «Методы синтеза широкополосных согласующих устройств к изменяющемуся импедансу нагрузки»	30
Железнёв А. И., Хаджинов М. К. «Анализ качества покрытия технологического участка вакуумной установкой»	32
Бейсенби М. А., Исатаева Г. С., Марков А. В. «Исследование робастной устойчивости системы по одному параметру в классе катастрофы «гиперболическая омбилика»	34
Калоша С. О., Нестеренков С. Н. «Технология Ethernet over USB как основа вычислительного кластера на Raspberry Pi»	36
Капцевич О. А., Рабченко Д. И. «Определение необходимых элементов пользовательского интерфейса и последовательности их размещения в автоматизированной системе управления»	38
Кардаш С. Н. «Ортогонализация системы ДНФ булевых функций»	40
Набешко Г. А., Захарьев В. А. «Стандарты токенов сети Ethereum и их особенности в реализации работы смарт-контрактов»	42
Навросюк В. Г. «Автоматизация разработки порталов с многократно используемыми компонентами»	44
Парамонова А. Е., Архипенко Я. С., Захарьев В. А. «Анализ платформ построения естественно-языковых интерфейсов»	46
Петровский И. И., Свито И. Л. «Применение элементов высшего порядка в фильтрах нижних и высших частот»	48
Русак Л. В., Стасевич Н. А. «Анализ дискретных систем методом уравнений моментов»	52
Снисаренко С. В. «Методы планирования траектории в неизвестной среде»	54
Третьяков А. С. «Разработка программного обеспечения IM View для исследования тепловентиляционных режимов работы асинхронных электродвигателей»	56

Хаджинов М. К., Доманов А. Т., Павлова А. В. «Сервопривод с нелинейным комбинированным управлением»	58
Шилин Л. Ю., Кукин Д. П. «Параметрическое проектирование импульсных систем фазовой синхронизации»	60
Яскевич Д. Н., Довнар А. Д., Курулёв А. П. «Обзор схемотехнических реализаций систем управления аккумуляторных батарей»	62
Секция «Интеллектуальные информационные технологии» ...	65
Boris Assanovich, Yury Veretilo «Deep Features Creation for Smile Classification in Biometric Systems»	66
Войтиховский П. Д., Сафоненко К. А., Давыденко И. Т. «Адаптивная система обучения английскому языку на основе кластерного анализа»	68
Гунер М. В., Бусс И. А. «Проектирование автоматизированной информационной системы управления коровной фермой с блоком идентификации животного и распознавания уровня его активности по видео с web-камеры»	70
Жук А. А. «Формулировка задачи оптимизации приложений, работающих с большими данными»	72
Зязюлькин С. П., Нестеренков С. Н. «Использование actor-critic алгоритмов при обучении агентов для игр на ATARI 2600»	74
Ивашенко В. П. «Исследование производительности реализации параллельной редукционной схемы теоретико-множественных операций»	76
Ивашенко В. П. «Операционная семантика многоагентных систем обработки знаний»	78
Ивашенко В. П. «Справочно-проверяющая система на основе унифицированного семантического представления знаний»	80
Карпук А. А., Говорко А. В. «Построение онтологии предметной области информационной системы управления использованием радиочастотного спектра»	82
Курочкин А. В. «Обобщенная модель медицинских экспертных систем поддержки принятия решений»	84
Синельников П. М., Давыденко И. Т. «Средства организации и управления коллективным проектированием баз знаний»	86
Хлопцев А. А., Кузьма Ю. В. «Генетический алгоритм для оптимизации подбора параметров торговых ботов»	88
Черников И. А. «Онтологическая модель поддержки подготовки и проведения мероприятий»	90
Шангитова Ж. Е., Оспанова Т. Т., Марков А. В. «Применение искусственных нейронных сетей в химической промышленности»	92
Секция «Проектирование встраиваемых систем»	95
Бибило П. Н., Романов В. И. «Экспериментальное сравнение эффективности разложений Шеннона и Давио при синтезе логических схем»	96
Заливако С. С., Иванюк А. А. «Использование флеш-памяти в качестве источника случайности для реализации физически неклонировуемых функций»	98
Карташов А. А., Журавлёв В. И. «Оптимизация энергоэффективности микроконтроллеров для схем со сверхнизким энергопотреблением»	100
Качинский М. В., Петровский Н. А., Станкевич А. В. «Организация и пропускная способность интерфейса PCI-e ускорительной платы Alveo U250 с управляющим компьютером»	102

Кириенко Н. А. «Оптимизация функциональных описаний логических схем с использованием инверсных представлений»	104
Кохновский С. И., Иванюк А. А. «Оценка ключей комбинационного устройства всевозможных перестановок»	106
Логинова И. П. «Использование технологии виртуализации при параллельных вычислениях на системах с общей памятью»	108
Оверченко А. С. «Оптимизация аппаратно-программного средства сшивки изображений» ...	110
Черемисинова Л. Д., Черемисинов Д. И. «Проверка эквивалентности схем на транзисторном уровне»	112
Шамына А. Ю., Иванюк А. А. «Исследование характеристик физически неклонированной функции типа арбитр на платах быстрого прототипирования»	114
Шинкевич Н. Н. «Исследование безопасности бесконтактных смарт-карт типа MIFARE Classic»	116
Секция «Математическое моделирование и компьютерная графика»	119
Барановский И. В., Тернов Е. В., Башко Ю. А. «Моделирование энергетической и экономической эффективности тепловой модернизации жилого здания»	120
Вельков Д. Е., Фролов Я. И., Гуринович А. Б. «Риски генерации паролей»	122
Гранько С. В., Короткевич А. В. «Моделирование и расчет пространственного распределения концентрации ионно-имплантированной смеси»	124
Дорошко О. В., Сталевская С. Н. «Оценка параметров модели INAR(1)»	126
Дунин А. Т. «Процедурная генерация уровней с использованием двоичного разбиения пространства»	128
Заерко Д. В., Лишницкий В. А. «Применение алгоритма весового определения граничных пикселей для трехканальной модели передачи цвета»	130
Колдаева Е. Ю. «Математика игрового баланса на примере инкрементальных игр»	132
Кукин Д. П., Гриневич Я. Г., Шатилова О. О. «Моделирование генератора хаоса на базе каскадносвязанных систем фазовой синхронизации»	134
Пилипчук Л. А., Полячок Е. Н. «К методам построения начальных допустимых решений экстремальных задач сетевой оптимизации»	136
Пилипчук Л. А., Полячок Е. Н., Лобко А. Э., Шкурский Д. А. «О численных методах и технологиях построения кратчайших путей»	138
Потапчик В. С. «Тестирование интерфейса мобильных игр»	140
Рак Т. А., Гриневич Я. Г. «Рендеринг реалистичных объектов на основе метода трассировки лучей»	142
Савенко А. Г. «Визуализатор трёхмерных моделей для реализации виртуальных лабораторных и практических занятий»	144
Цегельник В. В. «Об одном свойстве хаотических потоков с кривой равновесия»	146
Секция «Системы обработки информации»	148
Азаренко А. В. «Обзор методов статического анализа программного кода»	149
Белошедов Е. С., Гуринович А. Б., Гаруля Д. В., Архипенко Я. С. «Алгоритмы стохастического градиентного спуска обучения и тренировки нейронных сетей»	151

Бессараб З. И. «Выбор оптимальной облачной платформы для организации хранилища и обработки данных»	153
Бобков А. А. «Методы получения и обработки данных в системах печати документов на основе переменных данных»	155
Бранцевич П. Ю., Лапицкая Н. В., Леванцевич В. А. «Использование преобразования Гильберта-Хуанга для локализации кратковременных возмущений вибросигналов»	157
Бухаров Д. С. «О методе настройки устройства АЛАР КПАМ на основе равномерных данных»	159
Бык О. В. «Оптимальные алгоритмы прогнозирования бизнес-процессов»	161
Гаруля Д. В., Навроцкий А. А., Белощедов Е. С. «Методы оптимизации на основе градиентного спуска с моментом в задачах обучения нейронных сетей»	163
German J. O. «One version of the group resolution principle for discrete optimization»	165
Гуменников Е. Д., Мурашко И. А. «Генератор эквивалентного текста на базе LSTM сети» 167	
Жиляк Н. А. «Особенности архитектурной организации Вычислительных систем реального времени. Интегрированные конвейерные вычислители»	169
Кожемякин Е. А., Дубяга С. П., Курулёв А. П. «Способы кодирования состояний цифровых автоматов»	171
Козарь Р. В. «Алгоритмы распознавания медицинских изображений в задачах компьютерной автоматизированной диагностики»	173
Кузнецов М. В., Герман О. В. «Управление динамикой сложной системы»	175
Кузьма Ю. В., Хлопцев А. А. «Алгоритмы и методы машинного обучения для оценки устойчивости физически неклонируемых функций к криптографическим атакам»	177
Лабоха А. К., Шамына А. Ю., Ардяко А. Д. «Применение вегетационных индексов в системе мониторинга пожароопасной обстановки в лесах»	179
Леонович А. С., Ярмолик С. Н., Свинарский М. В., Храменков А. С. «Задача оценивания координат точек падения артиллерийских снарядов при обслуживании стрельбы» .	181
Минковский Д. В. «Применение платформы SAP HANA для анализа больших объёмов данных»	183
Мазура А. А., Гуринович А. Б. «Алгоритмы оптимизации сборки web-приложений»	185
Мазура И. А., Гуринович А. Б. «Алгоритм извлечения ключевых слов»	187
Минько Д. А. «Алгоритм улучшения качества медиафайлов»	189
Момотова Ю. О. «Анализ этапов предобработки и сегментации медицинских изображений больных COVID-19»	191
Мойсюк-Дранько П. А., Ревотюк М. П. «Методы матричной факторизации для систем рекомендации»	193
Mukha V. S., Kako N. F. «Flat problem of allowance distribution as dual control problem»	195
Пархименко В. А., Быков А. А. «Анализ мировой экономики на базе таблиц «затраты – выпуск» WIOD и Eora средствами языка R»	197
Ревотюк М. П., Хаджинова Н. В., Кузнецова О. В. «Встречный поиск кратчайших путей на динамически определяемых графах»	199
Савёнок В. А. «Уточнение динамической структуры из конечных автоматов для решения задачи поиска шаблона в тексте»	201
Савич С. И., Гуринович А. Б. «Алгоритм управления бизнес-процессами»	203

Удовин И. А., Воронова В. В. «Разработка тестирующей системы с использованием современных технологий изоляции процессов»	205
Хачатрян А. Г. «Деревья решений как метод диагностики злокачественных новообразований»	207
Черемисинова Л. Д., Черемисинов Д. И. «Верификация логических КМОП схем, устойчивых к мягким отказам»	209
Чигирь И. В. «Алгоритм совместного обнаружения - измерения параметров протяженной по дальности коррелированной ОШП в радиолокаторах сопровождения»	211
Шурмилов П. К. «Структуры пространственных данных в геоинформационных системах» ..	213
Ярмолик С. Н., Зайко Е. В., Свинарский М. В., Леонович А. С. «Адаптация к пространственной ориентации объектов в задаче сигнально-траекторного распознавания классов»	215

Научное издание

**Информационные технологии и системы 2020
(ИТС 2020)**

Материалы международной научной конференции
(БГУИР, Минск, Беларусь, 18 ноября 2020)

**Information Technologies and Systems 2020
(ITS 2020)**

Proceeding of The International Conference
(BSUIR, Minsk, Belarus, 18th November 2020)

В авторской редакции

Ответственный за выпуск *В.И. Журавлев, А.Б. Гуринович*

Компьютерная верстка *Я. Г. Гриневич, А. Ф. Трофимович*

Дизайн обложки *А. А. Навроцкий*

Подписано в печать 16.10.2013. Формат 60x84 1/8. Бумага офсетная. Гарнитура «Book Antiqua».
Отпечатано на ризографе. Усл. печ. л. 41,15. Уч.-изд. л. 45,3. Тираж 150 экз. Заказ 391.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6.



01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20

ITS 2020

iTechArt

software
technologies