

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

**Научный журнал Российской академии наук
(издается под руководством Отделения нанотехнологий
и информационных технологий РАН)**

Издается с 1989 года
Журнал выходит ежеквартально

Учредители:
Российская академия наук
Федеральный исследовательский центр
«Информатика и управление» Российской академии наук

РЕДАКЦИОННЫЙ СОВЕТ

академик РАН И. А. Соколов — председатель Редакционного совета
академик РАН Г. И. Савин
академик РАН А. Л. Степковский
член-корреспондент РАН Ю. Б. Зубарев
профессор Ш. Долев (S. Dolev, Beer-Sheva, Israel)
профессор Ю. Кабанов (Yu. Kabanov, Besancon, France)
профессор М. Никулин (M. Nikulin, Bordeaux, France)
профессор В. Ротарь (V. Rotar, San-Diego, USA)
профессор М. Финкельштейн (M. Finkelstein, Rostok, Germany)

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

академик РАН И. А. Соколов — главный редактор
профессор, д.ф.-м.н. С. Я. Шоргин — заместитель главного редактора
д.т.н. В. Н. Захаров проф., д.ф.-м.н. В. Ю. Королев
проф., д.т.н. В. Д. Ильин проф., д.г.-м.н. Р. Б. Сейфуль-Мулюков
проф., д.ф.-м.н. Л. А. Калиниченко проф., д.т.н. И. Н. Синицын
д.т.н. В. А. Козимиади к.т.н. А. В. Филин
проф., д.т.н. К. К. Колин к.ф.-м.н. С. А. Христочевский

Редакция

профессор, д.г.-м.н. Р. Б. Сейфуль-Мулюков
к.ф.-м.н. Е. Н. Арутюнов
С. Н. Стригина (ответственный секретарь)

© Федеральный исследовательский центр «Информатика
и управление» Российской академии наук, 2016

Журнал включен в базу данных Russian Science Citation Index (RSCI),
интегрированную с Web of Science

Журнал входит в систему Российского индекса научного цитирования (РИНЦ)
Журнал включен в базу данных CrossRef (систему DOI — Digital Object Identifier),
в базу данных Ulrich's periodicals directory
и в информационную систему «Общероссийский математический портал Math-Net.Ru»

Журнал реферируется в «Реферативном журнале» ВИНТИ
и в системе Google Scholar

Журнал включен в сформированный Минобрнауки России Перечень рецензируемых научных
изданий, в которых должны быть опубликованы основные научные результаты диссертаций
на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук

<http://www.ipiran.ru/journal/collected>

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 26 № 2 Год 2016

СОДЕРЖАНИЕ

Системы и средства глубокого обучения в задачах классификации

О. Ю. Бахтеев, М. С. Попова, В. В. Стрижов 4

Анализ самосинхронности электронных схем на нижнем уровне иерархии

Л. П. Плеханов 23

Лексический анализ динамически формируемых строковых выражений

М. И. Полубелова, С. В. Григорьев 43

Нормальные фильтры Пугачёва для автокоррелированных стохастических систем, линейных относительно состояния

И. Н. Синицын, Э. Р. Корепанов 63

Модифицированные эллипсоидальные субоптимальные фильтры для нелинейных стохастических систем на многообразиях

И. Н. Синицын, В. И. Синицын, Э. Р. Корепанов 79

Создание реалистичных наборов данных для алгоритмов трехмерной реконструкции с помощью виртуальной съемки компьютерной модели

О. А. Яковлев, А. В. Гасилов 98

Иерархическая форма представления биографического факта

И. М. Адамович, О. И. Волков 108

Информационная система ячеично-нейросетевого моделирования последствий химических аварий на опасных производственных объектах

С. П. Дударов, П. Л. Папаев 123

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 26 № 2 Год 2016

СОДЕРЖАНИЕ

Методы построения схемотехнических систем
автоматизированного проектирования с использованием
сервис-ориентированного подхода на базе протокола
WebSocket

В. И. Анисимов, В. Н. Гридин, С. А. Васильев **136**

Нормализация банковских онлайн-сервисов

В. Д. Ильин **147**

Когнитивные аспекты информационного моделирования
при проектировании сложных информационных систем

Г. В. Лукьянов, Д. А. Никишин **158**

Информационные технологии и образование в Евросоюзе:
формирование новых профессионалов

Д. А. Богданова **171**

Об авторах **181**

Правила подготовки рукописей статей **183**

Requirements for manuscripts **187**

СИСТЕМЫ И СРЕДСТВА ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧАХ КЛАССИФИКАЦИИ*

O. Ю. Бахтеев¹, М. С. Попова², В. В. Стрижов³

Аннотация: Работа посвящена построению сети глубокого обучения и оптимизации ее параметров с помощью вычислительных мощностей графического ускорителя на основе сервиса облачных вычислений Amazon Web Services (AWS). Рассматривается задача классификации временных рядов. Для ее решения строится сеть глубокого обучения: суперпозиция универсальных моделей. В качестве исследуемой структуры сети рассматривается композиция ограниченной машины Больцмана, автокодировщика и двухслойной нейросети. Анализируется зависимость ошибки классификации от числа параметров и размера обучающей выборки. В качестве иллюстрирующего примера рассматривается задача классификации временных рядов акселерометра мобильного телефона. Приведены рекомендации по использованию программного обеспечения, предназначенного для решения задач глубокого обучения.

Ключевые слова: классификация временных рядов; глубокое обучение; суперпозиция моделей; автокодировщик; ограниченная машина Больцмана; Theano; Amazon Web Services

DOI: 10.14357/08696527160201

1 Введение

В данной работе рассматривается задача построения сети глубокого обучения с использованием средств библиотеки NVIDIA CUDA и сервиса облачных вычислений AWS. Решается задача классификации временных рядов, где под временным рядом понимается упорядоченный по времени набор изменения некоторой случайной величины. Для решения этой задачи используются методы глубокого обучения. Под глубоким обучением понимается область машинного обучения, посвященная построению нелинейных моделей классификации или регрессии, элементы суперпозиции которых описывают соответствующий уровень признакового агрегирования данных [1].

*Работа выполнена при поддержке РФФИ (проект 16-37-00488).

¹Московский физико-технический институт, bakhteev@phystech.edu

²Московский физико-технический институт, maria.popova@phystech.edu

³Вычислительный центр им. А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, strijov@ccas.ru

Решению прикладных задач методами глубокого обучения посвящено значительное число современных работ. Работы [2–4] посвящены классификации временных рядов с использованием методов глубокого обучения. В работе [3] используются рекуррентные нейронные сети, т. е. сети, содержащие циклические связи [1]. В работе [4] для классификации временных рядов рассматриваются различные комбинации ограниченной машины Больцмана, автокодировщика и двухслойной нейронной сети. Исследуется суперпозиция, состоящая из ограниченной машины Больцмана, автокодировщика и двухслойной нейронной сети [1].

Работы [5–8] посвящены регуляризации сетей глубокого обучения. В работах [9, 10] рассматривается проблема неустойчивости сети. В работе [10] исследуется поведение модели глубокого обучения как липшицевой функции. Работы [11–13] посвящены ускорению процесса решения оптимизационных задач, возникающих при работе с сетями глубокого обучения. Работы [9, 14] посвящены построению адаптивных моделей глубокого обучения. Работа [15] посвящена рекуррентной модификации модели ограниченной машины Больцмана [16] для классификации временных рядов. Схожие идеи предлагаются в работе [17], посвященной построению рекурсивного автокодировщика.

В работе [18] рассматривается регуляризация автокодировщика, позволяющая регулировать разреженность внутреннего слоя автокодировщика. В работе [19] рассматривается принцип работы варианта автокодировщика, задачей которого является фильтрация шума входного сигнала.

В данной статье решается прикладная задача классификации временных рядов. В качестве данных для вычислительного эксперимента используются данные с акселерометров мобильных телефонов [20]. Эксперимент проводится с использованием Theano [21, 22] — библиотеки для вычислений для языка Python. Theano активно используется для построения моделей глубокого обучения, а также для построения более специализированных библиотек [23, 24]. Функции вычислений Theano компилируются, что позволяет выполнять вычисления за приемлемое время. Другой отличительной особенностью Theano является возможность применения графического процессора с использованием архитектуры параллельных вычислений CUDA [25]. Также в экспериментальном режиме доступна возможность применить интерфейс OpenCL [26], что позволяет использовать для обучения нейросетей графические процессоры не только NVIDIA, но и других производителей, например Xilinx. В данной работе описывается эксперимент, направленный на сравнение эффективности обучения искусственных нейронных сетей на центральном процессоре и на графическом процессоре компьютера. Приводятся рекомендации по технической реализации классификации с использованием средств сервиса облачных вычислений AWS. Для решения задачи оптимизации используется алгоритм обратного распространения ошибок с послойным предобучением сети и дальнейшей настройкой параметров всех слоев [27]. В качестве альтернативного инструмента для решения задачи с использованием глубокого обучения рассматривается программное обеспечение для решения задач классификации временных рядов на языке Matlab.

2 Формальная постановка задачи

Пусть задана выборка

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}, \quad i = 1, \dots, N, \quad (1)$$

состоящая из множества пар «объект–класс», $\mathbf{x}_i \in \mathbb{R}^n$. Каждый объект \mathbf{x} принадлежит одному из Z классов с меткой $y_i \in \mathbf{Y} = \{1, \dots, Z\}$.

Моделью классификации или сетью глубокого обучения \mathbf{f} назовем суперпозицию функций

$$\mathbf{f}(\mathbf{w}, \mathbf{x}) = \boldsymbol{\mu}_1(\boldsymbol{\mu}_2(\dots \boldsymbol{\mu}_K(\mathbf{x}))) : \mathbb{R}^n \rightarrow [0, 1]^Z, \quad (2)$$

где $\boldsymbol{\mu}_k, k \in \{1, \dots, K\}$, — модели, параметрическое семейство вектор-функций; \mathbf{w} — вектор параметров моделей; r -ю компоненту вектора $\mathbf{f}(\mathbf{x}, \mathbf{w})$ будем интерпретировать как вероятность отнесения объекта \mathbf{x}_i к классу с меткой r .

Требуется минимизировать функцию ошибки S на обучающей выборке \mathfrak{D} , где S — сумма отрицательных логарифмов правдоподобия [28] по всем объектам выборки

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} S(\mathbf{w} | \mathfrak{D}),$$

где

$$S(\mathbf{w} | \mathfrak{D}) = - \sum_{(\mathbf{x}, y) \in \mathfrak{D}} \sum_{r=1}^Z [y_i = r] \log p(y = r | \mathbf{x}, \mathbf{w}).$$

2.1 Структура сети глубокого обучения

Предлагается использовать в качестве алгоритма решения задачи суперпозицию, состоящую из трех основных компонент: ограниченной машины Больцмана, автокодировщика и двухслойной нейросети с softmax-классификатором.

Ограниченнная машина Больцмана. Ограниченнная машина Больцмана представляет собой двудольный граф, где первая доля соответствует переменной \mathbf{x} , а вторая доля — бинарному вектору \mathbf{h} длины n' . Рассмотрим случай, когда вектор \mathbf{x} принимает бинарные значения. Определим энергию пары входного слоя \mathbf{x} и скрытого слоя \mathbf{h} следующим образом:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{x}^T \cdot \mathbf{b}_{\text{vis}} - \mathbf{h}^T \cdot \mathbf{b}_{\text{hid}} - \mathbf{h}^T \mathbf{W}_{\text{RBM}} \mathbf{x},$$

где \mathbf{b}_{vis} , \mathbf{b}_{hid} и \mathbf{W}_{RBM} — параметры модели.

Пусть совместное распределение пары векторов \mathbf{x} , \mathbf{h} задано следующим образом:

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{x}, \mathbf{h})),$$

где Z — нормировочный коэффициент:

$$Z = \sum_{\mathbf{x} \in \{0,1\}^n, \mathbf{h} \in \{0,1\}^{n'}} \exp(-E(\mathbf{x}, \mathbf{h})) .$$

Функция вероятностей вектора \mathbf{x} есть сумма вероятностей по всем скрытым состояниям вектора \mathbf{h} :

$$p(\mathbf{x}) = \sum_{\mathbf{h} \in \{0,1\}^{n'}} p(\mathbf{x}, \mathbf{h}) .$$

Определим элемент суперпозиции (2):

$$\mu_{\text{RBM}}(\mathbf{x}) = E(\mathbf{h}|\mathbf{x}) . \quad (3)$$

Настройка параметров модели (3) осуществляется решением задачи оптимизации:

$$\begin{aligned} \hat{\mathbf{W}}_{\text{RBM}}, \hat{\mathbf{b}}_{\text{vis}}, \hat{\mathbf{b}}_{\text{hid}} &= \arg \max_{\mathbf{W}_{\text{RBM}}, \mathbf{b}_{\text{vis}}, \mathbf{b}_{\text{hid}}} p(\mathfrak{D}; \mathbf{W}, \mathbf{b}_{\text{vis}}, \mathbf{b}_{\text{hid}}) = \\ &= \prod_{\mathbf{x} \in \mathfrak{D}} \sum_{\mathbf{h} \in \{0,1\}^{n'}} \frac{1}{Z} \exp(-E(\mathbf{x}, \mathbf{h})) . \end{aligned} \quad (4)$$

В данной работе используется модифицированная версия ограниченной машины Больцмана, позволяющая работать с небинарными входными данными [29]. В этой модификации энергия E пары входного слоя \mathbf{x} и скрытого слоя \mathbf{h} выглядит следующим образом:

$$E(\mathbf{x}, \mathbf{h}) = \frac{(\mathbf{x} - \mathbf{b}_{\text{vis}})^2}{2\sigma^2} - \mathbf{h}^T \cdot \mathbf{b}_{\text{hid}} - \frac{\mathbf{h}^T}{\sigma} \mathbf{W} \mathbf{x} ,$$

где σ — стандартное нормальное отклонение объектов выборки \mathfrak{D} , деление производится покомпонентно.

Для решения задачи оптимизации (4) используется алгоритм, описанный в [1].

Автокодировщик. Автокодировщик предназначен для снижения размерности исходного пространства признаков. Автокодировщик представляет собой суперпозицию кодирующего и декодирующего блока [1]:

$$\mu'_{\text{AE}} = \phi(\mathbf{g}(\mathbf{x})) ,$$

где $\mathbf{g}(\mathbf{x}) = \sigma(\mathbf{W}_e \mathbf{x} + \mathbf{b}_e)$ — кодирующий блок; $\phi(\mathbf{g}(\mathbf{x})) = \sigma(\mathbf{W}_d \mathbf{g}(\mathbf{x}) + \mathbf{b}_d)$ — декодирующий блок; $\sigma(t) = (1 + \exp(-t))^{-1}$ — сигмоидная функция; \mathbf{W}_e , \mathbf{W}_d , \mathbf{b}_e и \mathbf{b}_d — параметры модели.

Введем дополнительное ограничение на матрицы \mathbf{W}_e и \mathbf{W}_d :

$$\mathbf{W}_e = \mathbf{W}_d^T .$$

Оптимизацию параметров модели \mathbf{W}_e будем проводить таким образом, чтобы по образу вектора \mathbf{x} , получаемому с помощью кодирующего блока, можно было получить вектор $\boldsymbol{\mu}_{AE}$, близкий к исходному входному \mathbf{x} , при помощи преобразования декодирующего блока:

$$\hat{\mathbf{W}}_e, \hat{\mathbf{W}}_d, \hat{\mathbf{b}}_e, \hat{\mathbf{b}}_d = \arg \min_{\mathbf{W}_e, \mathbf{W}_d, \mathbf{b}_e, \mathbf{b}_d} \frac{1}{|\mathfrak{D}|} \sum_{\mathbf{x} \in \mathfrak{D}} \|\boldsymbol{\mu}_{AE}(\mathbf{x}) - \mathbf{x}\|_2^2. \quad (5)$$

Стоит отметить, что декодирующий блок ϕ требуется только для решения задачи оптимизации (5) и не используется в суперпозиции (2). Таким образом, элемент суперпозиции (2) определен как

$$\boldsymbol{\mu}_{AE} = \mathbf{g}(\mathbf{x}).$$

Двухслойная нейросеть. Двухслойная сеть представляет собой логистическую вектор-функцию [30]:

$$\left. \begin{aligned} \mathbf{a}(\mathbf{x}) &= \mathbf{W}_2^\top \tanh(\mathbf{W}_1^\top \mathbf{x}) ; \\ \boldsymbol{\mu}_{SM}(\mathbf{x}) &= \frac{\exp(\mathbf{a}(\mathbf{x}))}{\sum_{j=1}^Z \exp(a_j(\mathbf{x}))} , \end{aligned} \right\} \quad (6)$$

где r -я компонента вектора $\boldsymbol{\mu}_{SM}(\mathbf{x})$ интерпретируется как вероятность принадлежности объекта \mathbf{x} классу r . Итоговая функция классификации (2) ставит в соответствие объекту \mathbf{x} метку класса y , где y — класс, к которому принадлежит \mathbf{x} с наибольшей вероятностью:

$$f(\mathbf{w}, \mathbf{x})(r) = \begin{cases} 1, & \text{если } r = \arg \max_{r'} \boldsymbol{\mu}_{SM}(\boldsymbol{\mu}_{AE}(\boldsymbol{\mu}_{RBM}(\mathbf{x}))(r')) ; \\ 0 & \text{иначе.} \end{cases}$$

Здесь $\boldsymbol{\mu}_{AE}$ и $\boldsymbol{\mu}_{RBM}$ — автокодировщик (5) и ограниченная машина Больцмана (4) соответственно; $\boldsymbol{\mu}_{SM}(\mathbf{x})(r)$ — r -я компонента вектора $\boldsymbol{\mu}_{SM}$; $f(\mathbf{w}, \mathbf{x})(r)$ — r -я компонента вектор-функции \mathbf{f} .

Итоговая задача оптимизации выглядит следующим образом:

$$\hat{\Theta} = \arg \min_{\mathbf{x}, y \in \mathfrak{D}} \sum_{r=1}^Z [y = r] \log (\boldsymbol{\mu}_{SM}^r (\boldsymbol{\mu}_{AE} (\boldsymbol{\mu}_{RBM} (\mathbf{x})))) ,$$

где $\hat{\Theta} = [\hat{\mathbf{W}}_{RBM}, \hat{\mathbf{b}}_{vis}, \hat{\mathbf{b}}_{hid}, \hat{\mathbf{W}}_e, \hat{\mathbf{b}}_e, \hat{\mathbf{W}}_2^\top, \hat{\mathbf{W}}_1^\top]$ — параметры ограниченной машины Больцмана (4), автокодировщика (5) и двухслойной сети (6).

3 Запуск эксперимента на Amazon Web Services

В данном разделе кратко изложен порядок действий по запуску вычислительного эксперимента для решения задачи (2) на платформе AWS.

Необходимое программное обеспечение. Для запуска алгоритма на AWS требуется предварительно зарегистрироваться и активировать аккаунт на сайте AWS. Также следует установить утилиты PUTTY, PUTTYGen и WinSCP для Windows, либо scp и ssh для Linux и Mac OS.

Настройка экземпляра (instance). Под экземпляром (*англ. instance*) AWS понимается предоставляемая сервисом вычислительная платформа, с которой можно работать как с удаленным сервером. После авторизации на странице профиля AWS следует зайти на консоль управления [31] и выбрать пункт «EC2», ведущий в подраздел «EC2 Dashboard». Для настройки нового экземпляра машины AWS следует нажать кнопку «Launch Instance» из этого подраздела. В новом окне будет доступен выбор операционной системы. Для запуска вычислительного эксперимента на Theano наиболее простым вариантом является выбор операционной системы, сконфигурированной сообществом для работы с CUDA и Theano, перечень которых можно найти на вкладке «Community AMIs». Необходимо зайти во вкладку «Community AMIs», в строке поиска ввести имя конфигурации и нажать кнопку «Select». В данной работе использовалась конфигурация «`ami-1797eb27`», являющаяся версией операционной системы Ubuntu с предустановленным пакетом Theano (рис. 1).

После выбора конфигурации операционной системы требуется выбрать тип экземпляра и его характеристики: объем жесткого диска, объем оперативной памяти, скорость сети, наличие графического ускорителя и т. п. В данной работе используется платный тип экземпляра `g2.2xlarge`, обладающий восьмиядер-

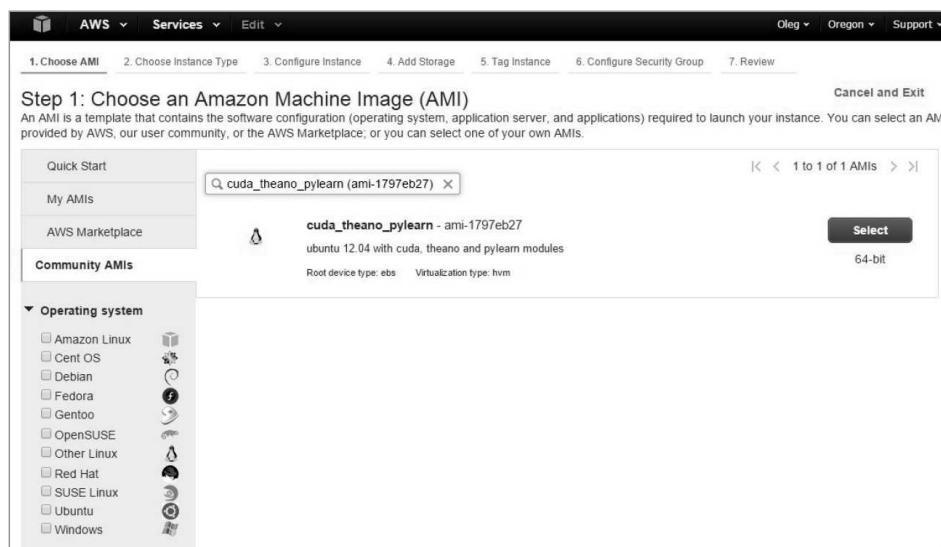


Рис. 1 Выбор конфигурации на консоли AWS

ным процессором, 15 ГБ памяти и поддержкой графического ускорителя NVIDIA GRID. В случае если не требуется поддержки графического процессора, можно использовать бесплатный тип экземпляра **t2.micro**. Дальнейшие настройки экземпляра опциональны. После завершения настройки система предложит создать пару из открытого и закрытого ключа для авторизации в операционной системе экземпляра или выбрать существующий. После завершения настройки экземпляра новый экземпляр появится в таблице на вкладке «**Instances**», доступной в разделе «**EC2 Dashboard**». Там же будет доступен и IP-адрес экземпляра (поле «**Public IP**»).

Доступ к экземпляру. При работе с экземпляром из операционных систем Linux и Mac OS требуется изменить права для файла ключа:

```
chmod 400 <путь к ключу>
```

Доступ к экземпляру можно получить по протоколу SSH по адресу, указанному системой после создания экземпляра. Для доступа из операционной системы Linux и Mac OS требуется ввести на консоли строку:

```
ssh -i <путь к ключу> ubuntu@<ip-адрес>
```

Для доступа из операционной системы Windows можно использовать программу PuTTY и PuTTYgen. PuTTYgen конвертирует закрытый ключ в приемлемый для AWS вид. Для конвертации ключа требуется выбрать в программе

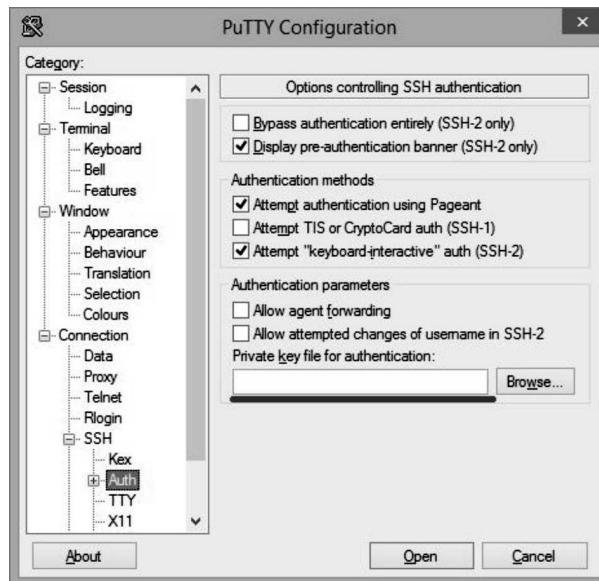


Рис. 2 Настройка аутентификации Putty

пункт File-Load private key и выбрать полученный от AWS ключ с расширением pem. Затем требуется выбрать кнопку «Save private key» и сохранить полученный файл с расширением ppk.

Для дальнейшего использования ключа в Putty требуется выбрать вкладку Connection-SSH-Auth, нажать кнопку «Browse» и выбрать ppk-ключ (рис. 2). После этого можно заходить на сервер, указав в поле «Host name» строку вида

```
ubuntu@<ip-адрес>
```

Обмен файлами с экземпляром. Для обмена файлами с экземпляром можно использовать утилиту scp для Linux и Mac OS и WinSCP для Windows. Как и в случае с соединением по SSH (Secure SHell), при работе с экземпляром из операционных систем Linux и Mac OS требуется изменить права для файла ключа:

```
chmod 400 <путь к ключу>
```

Для копирования данных на сервер экземпляра с использованием утилиты scp требуется ввести на консоли строку

```
scp -i <ключ> <локальный путь к файлу> ubuntu@<ip-адрес>:  
<путь к файлу на экземпляре>
```

Для копирования данных с сервера экземпляра на локальный компьютер с использованием утилиты scp требуется ввести на консоли строку

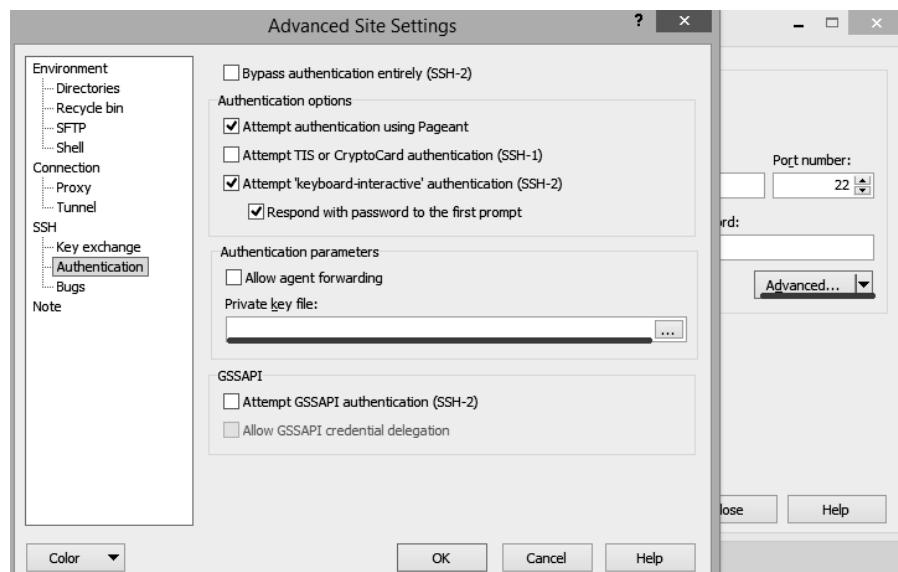


Рис. 3 Настройка аутентификации WinSCP

```
scp -i <путь к ключу> ubuntu@<ip-адрес>:  
<путь к файлу на экземпляре> <локальный путь к файлу>
```

В случае копирования папки требуется добавить в команде параметр «-r»:

```
scp -i <ключ> -r <локальный путь к папке>  
ubuntu@<ip-адрес>:<путь к папке на экземпляре>
```

В операционной системе Windows для копирования данных на сервер экземпляра необходимо открыть программу WinSCP. В поле «Host name» нужно ввести IP-адрес «Public IP», а в поле «User name» — строку «ubuntu». Затем требуется нажать кнопку «Advanced», в новом окне выбрать вкладку «SSH» — «Authentification», нажать кнопку «...» и выбрать ppk-файл, сгенерированный PuTTYGen (рис. 3). Затем требуется нажать кнопку «Login». После этого можно передавать документы между локальным компьютером и экземпляром AWS.

Завершение работы с экземпляром. По завершении работы с экземпляром требуется уничтожить его, выбрав его на консоли управления и использовав команду «Terminate». **Важно:** на консоли также имеется команда «Stop», отключающая экземпляр. За хранение данных на отключенном экземпляре может взиматься дополнительная плата. Кроме того, в 2014 г. была зафиксирована ошибка AWS, взимавшая дополнительную плату за отключенные инстансы [32], поэтому рекомендуется удалять экземпляр полностью. Также рекомендуется удалить все хранилища данных, присоединенные к экземпляру (вкладка Volumes).

Запуск вычислительного эксперимента. Для запуска вычислительного эксперимента с использованием Theano требуется указать параметры конфигурации Theano перед запуском. В противном случае будет принята конфигурация по умолчанию. Наиболее простым вариантом установки параметров Theano является вызов программы с использованием следующего синтаксиса:

```
THEANO_FLAGS=<параметры Theano в формате «ключ=значение»  
через запятую> python <путь к скрипту>
```

Для запуска скрипта с n процессорами требуется вызвать его следующим образом:

```
OMP_NUM_THREADS=n python <путь к скрипту>
```

Для запуска скрипта с использованием CUDA требуется вызвать скрипт следующей командой:

```
THEANO_FLAGS=device=gpu,floatX=float32 python  
<путь к скрипту>
```

Здесь `floatX=float32` — параметр, отвечающий за формат вещественных чисел при обработке Theano. В настоящее время при работе с CUDA поддерживается только float32.

Общий план запуска алгоритма. Ниже приведена общая пошаговая инструкция запуска алгоритма на сервере AWS:

- зарегистрироваться на AWS;
- сконфигурировать экземпляр;
- скопировать код и данные проекта на сервер (например, в папку `/home/ubuntu/`);
- присоединиться к серверу по протоколу SSH;
- зайти в папку с проектом с использованием команды `cd`;
- запустить вычислительный эксперимент;
- скопировать результаты эксперимента на локальный компьютер;
- уничтожить экземпляра.

В качестве примера запуска вычислительного эксперимента можно вызвать команду `sh run8CPU.sh` для запуска эксперимента на восьми ядрах процессора или `sh runGPU.sh` для запуска эксперимента на графическом ускорителе. Подробно параметры запуска описаны в Readme-файле проекта [33].

4 Построение сети глубокого обучения заданной структуры на языке Matlab

В качестве независимого варианта решения задачи (2) предлагается использовать инструментарий [34] авторов [4] на языке Matlab. Он предназначен для построения нейронных сетей глубокого обучения со структурой, задаваемой пользователем.

Описание процедуры запуска инструментария. Для настройки и тестирования сети необходим набор данных. Данные должны быть сохранены в формате csv (comma separated values), а файл должен быть назван `dataset.csv`. Файл `dataset.csv` состоит из N числовых строк. Первое число в каждой строке файла должно быть целым от 1 до Z – это метка класса. Оставшиеся m чисел в строке — это числовые признаки объекта (1). Далее необходимо задать желаемую структуру модели. Для этого нужно создать файл `model.csv`. Этот файл состоит из одной числовой строки. Первое число — это желаемое число уровней суперпозиции модели. Далее обозначим как K . Следующие K чисел — описание типов уровней (1 — ограниченная машина Больцмана; 2 — автокодировщик; 3 — двухслойная нейросеть); следующие K чисел — число нейронов в каждом слое с первого по K -й соответственно; следующие K чисел — число циклов обучения для каждого слоя. Последнее число в строке — число объектов выборки, которые будут случайным образом выбраны из всего набора данных для обучения модели.

Пример. Стока 3 1 2 3 100 80 50 100 100 200 500 описывает следующую модель: трехслойная суперпозиция ограниченной машины Больцмана со 100 нейронами,

автокодировщика с 80 нейронами и двухслойной нейросети с 50 нейронами. Будет проведено 100 циклов обучения RBM, 100 циклов обучения автокодировщика и 200 циклов обучения двухслойной нейросети. Для обучения модели случайным образом будет выбрана подвыборка мощности 500.

Замечание. Ограниченные машины Больцмана и автокодировщики обучаются «без учителя» и задают процедуру порождения признаков, т. е. на выходе этих моделей получается преобразованная матрица признакового описания. Двухслойная нейросеть обучается «с учителем» и непосредственно разделяет объекты на классы. Учитывая это, структура K -слойной модели должна быть задана таким образом, чтобы первые $K - 1$ слоев состояли из машин Больцмана и автокодировщиков, а последний слой — из двухслойной нейросети.

Настройка нейронной сети. После подготовки файлов с набором данных и описанием суперпозиции можно создать, обучить и протестировать модель. Для этого необходимо выполнить следующие шаги:

- (1) загрузить на компьютер инструментарий из репозитория MLAlgorithms [34];
- (2) запустить систему Matlab;
- (3) в качестве рабочей директории в Matlab установить директорию:
... \Popova2015DeepLearning;
- (4) поместить файл dataset.csv в папку ... \Popova2015DeepLearning\data, а файл model.csv — в папку ... \Popova2015DeepLearning\code;
- (5) запустить файл main.m, который находится в папке:
... \Popova2015DeepLearning\code.

После того как файл main.m завершит работу, в папке ... \Popova2015DeepLearning\report будут сохранены графики с ROC-кривой (Receiver Operating Characteristic) и функционалом AUC (Area Under Curve), а также файл model.mat, в котором хранится настроенная модель.

5 Вычислительный эксперимент

В качестве данных для проведения вычислительного эксперимента использовались данные WISDM (WIreless Sensor Data Mining) [20], представляющие собой набор записей акселерометра мобильного телефона. Каждой записи соответствуют три координаты по осям акселерометра. Набор данных содержит записи движений для 6 классов переменной длины. При проведении вычислительного эксперимента из каждой записи использовались первые 200 сегментов. Так как выборка не сбалансирована, в нее добавлялись повторы записей классов, содержащих количество записей, меньшее чем у большего класса.

Основные эксперименты — исследование зависимости ошибки классификации от числа параметров и размера выборки — были проведены как с использованием инструментария на базе библиотеки Theano, так и с использованием

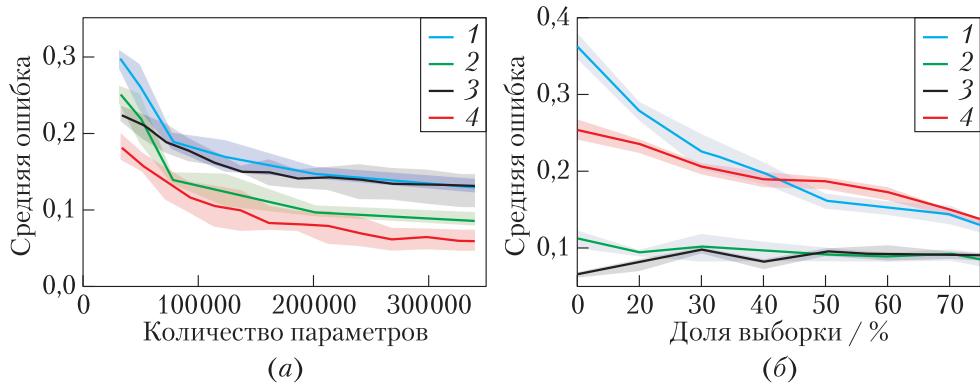


Рис. 4 Зависимость ошибки от числа нейронов (а) и от размера обучающей выборки (б): 1 — ошибка на контрольной выборке, Theano; 2 — ошибка на обучающей выборке, Theano; 3 — ошибка на контрольной выборке, Matlab; 4 — ошибка на обучающей выборке, Matlab

инструментария на языке Matlab. Для оценки качества классификации была проведена процедура скользящего контроля [35] при соотношении числа объектов обучающей и контрольной выборки 3 : 1. Число нейронов на каждом слое задавалось из соотношения 10 : 6 : 3. При проведении процедуры скользящего контроля для каждого отсчета количества нейронов было произведено пять запусков. В эксперименте с использованием инструментария на базе Theano при обучении двухслойной нейронной сети проводился мультистарт [36], т. е. одновременный запуск обучения сети с 8 разными стартовыми значениями параметров для предотвращения возможного застревания алгоритма обучения в локальном минимуме. При оценке качества классификации выбиралась модель с наилучшими результатами. График зависимости ошибки классификации от числа используемых нейронов изображен на рис. 4, а.

Для оценки зависимости качества классификации от размера обучающей выборки была проведена кроссвалидация с фиксированным количеством объектов в обучающей выборке (25% исходной выборки) и переменным размером обучающей выборки. Число нейронов было установлено как 364 : 224 : 112. При проведении процедуры скользящего контроля для каждого отсчета было произведено пять запусков. График зависимости ошибки классификации от размера обучающей выборки представлен на рис. 4, б.

Для исследования скорости работы процесса обучения нейросети в зависимости от конфигурации Theano был сделан следующий эксперимент: проводилось обучение двухслойной нейросети на основе подсчитанных заранее параметров ограниченной машины Больцмана (4) и автокодировщика (5). Обучение проходило за 100 итераций. При обучении алгоритм запускался параллельно с n разными

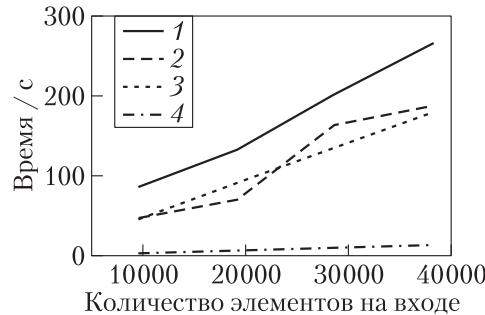


Рис. 5 Результаты эксперимента по исследованию скорости процесса обучения: 1 — CPU, 1 ядро; 2 — CPU, 4 ядра; 3 — CPU, 8 ядер; 4 — GPU

стартовыми позициями, $n \in \{1, \dots, 4\}$. Число нейронов было установлено как $300 : 200 : 100$. Запуск осуществлялся со следующими конфигурациями Theano:

- вычисление на центральном процессоре, задействовано одно ядро;
- вычисление на центральном процессоре, задействовано четыре ядра;
- вычисление на центральном процессоре, задействовано восемь ядер;
- вычисление на графическом процессоре.

Результаты эксперимента приведены на рис. 5. Как видно из графика, вычисление с использованием CUDA показывает значительное ускорение по сравнению с вычислением на центральном процессоре.

6 Заключение

В данной работе был проведен ряд вычислительных экспериментов с использованием библиотеки Theano и сервисом облачных вычислений AWS. Проведены эксперименты для установления зависимости ошибки классификации временных рядов с использованием сетей глубокого обучения от размера выборки и числа параметров сети. Исследовалась эффективность обучения искусственной нейросети с использованием графического процессора. Эксперимент показал значительное ускорение вычислений на графическом процессоре. Исходный код экспериментов доступен по адресам [33, 34].

Литература

1. Cho K. Foundations and advances in deep learning. — Espoo: Aalto University, 2014. DSc. Diss.
2. Langkvist M., Karlsson L., Loutfi A. A review of unsupervised feature learning for time-series modelling // Pattern Recogn. Lett., 2014. Vol. 42. No. 1. P. 11–24.

3. Desell T., Clachar S., Higgins J., Wild B. Evolving deep recurrent neural networks using ant colony optimization // Evolutionary computation in combinatorial optimization / Eds. G. Ochoa, F. Chicano. — Lecture notes in computer science ser. — Springer, 2015. Vol. 9026. P. 86–98.
4. Попова М. С., Стрижов В. В. Построение нейронных сетей глубокого обучения для классификации временных рядов // Системы и средства информатики, 2015. Т. 25. № 3. С. 60–77.
5. Wager S., Wang S., Liang P. Dropout training as adaptive regularization // Adv. Neur. In., 2013. Vol. 26. P. 351–359.
6. Wan L., Zeiler M., Zhang S., LeCun Y., Fergus R. Regularization of neural networks using DropConnect // 30th Conference (International) on Machine Learning Proceedings, 2013. P. 1058–1066.
7. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting // J. Mach. Learn. Res., 2014. Vol. 15. P. 1929–1958.
8. Gal Y., Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. arXiv.org, 2015. <http://arxiv.org/abs/1506.02142>.
9. Goodfellow I. J., Le Q. V., Saxe A. M., Lee H., Ng A. Y. Measuring invariances in deep networks // Adv. Neur. In., 2009. Vol. 22. P. 646–654.
10. Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I., Fergus R. Intriguing properties of neural networks. arXiv.org, 2014. <http://arxiv.org/abs/1312.6199>.
11. Raiko T., Valpola H., LeCun Y. Deep learning made easier by linear transformations in perceptrons // J. Mach. Learn. Res., 2012. Vol. 22. P. 924–932.
12. Bengio Y., Laufer E., Alain G., Yosinski G. Deep generative stochastic networks trainable by backprop // 31st Conference (International) on Machine Learning Proceedings, 2014. P. 226–234.
13. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv.org, 2015. <http://arxiv.org/abs/1502.03167>.
14. Li Z., Chang C., Liang F., Huang T. S., Cao C., Smith J. R. Learning locally-adaptive decision functions for person verification // IEEE Conference on Computer Vision and Pattern Recognition Proceedings, 2013. P. 3610–3617.
15. Sutskever I., Hinton G., Taylor G. The recurrent temporal restricted Boltzmann machine // Adv. Neur. In., 2009. Vol. 21. P. 1601–1608.
16. Fischer F., Igel C. Training restricted Boltzmann machines: An introduction // Pattern Recogn., 2014. Vol. 47. P. 25–39.
17. Socher R., Huang E. H., Pennington J., Ng A. Y., Manning C. D. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection // Adv. Neur. In., 2011. Vol. 24. P. 801–809.
18. Shu M., Fyshe F. Sparse autoencoders for word decoding from magnetoencephalography // 3rd NIPS Workshop on Machine Learning and Interpretation in NeuroImaging Proceedings, 2013. <http://www.cs.cmu.edu/~afyshe/papers/SparseAE.pdf>.
19. Vincent P., Larochelle H., Lajoie I., Bengio Y., Manzagol P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion // J. Mach. Learn. Res., 2013. Vol. 11. P. 3371–3408.

20. Kwapisz J. R., Weiss G. M., Moore S. Activity recognition using cell phone accelerometers // SIGKDD Explorations, 2010. Vol. 12. No. 2. P. 74–82.
21. Bergstra J., Breuleux O., Bastien F., Lamblin P., Pascanu R., Desjardins G., Turian J., Warde-Farley D., Bengio Y. Theano: A CPU and GPU math expression compiler // Conference on Python for Scientific Computing Proceedings, 2010. P. 3–11.
22. Bastien F., Lamblin P., Pascanu R., Bergstra J., Goodfellow I., Bergeron A., Bouchard N., Warde-Farley D., Bengio Y. Theano: New features and speed improvements // arXiv.org, 2012. <http://arxiv.org/pdf/1211.5590v1.pdf>.
23. Goodfellow I. J., Warde-Farley D., Lamblin P., Dumoulin V., Mirza M., Pascanu R., Bergstra J., Bastien F., Bengio Y. Pylearn2: A machine learning research library. arXiv.org, 2013. <http://arxiv.org/abs/1308.4214>.
24. Dieleman S., Schlüter J., Raffel C., et al. Lasagne: First release // Zenodo, August 13, 2015. <http://dx.doi.org/10.5281/zenodo.27878>.
25. Nickolls J., Buck I., Garland M., Skadron K. Scalable parallel programming with CUDA // ACM Queue, 2008. Vol. 6. No. 2. P. 40–53.
26. Stone J. E., Gohara D., Shi G. OpenCL: A parallel programming standard for heterogeneous computing systems // J. IEEE Design Test, 2010. Vol. 12. No. 10. P. 66–73.
27. Erhan D., Bengio Y., Courville A., Manzagol P., Vincent P. Why does unsupervised pre-training help deep learning? // J. Mach. Learn. Res., 2010. Vol. 11. P. 625–660.
28. Duan K., Keerthi S. S., Chu W., Shevade S. K., Poo A. N. Multi-category classification by soft-max combination of binary classifiers // 4th Workshop (International) on Multiple Classifier Systems, 2003. P. 125–134.
29. Cho K., Raiko T., Ilin A. Gaussian-Bernoulli deep Boltzmann machine // Joint Conference (International) on Neural Networks, 2013. P. 1–7.
30. Hinton G. E., Osindero S., Teh Y. A fast learning algorithm for deep belief nets // Neural Comput., 2006. Vol. 18. P. 1527–1554.
31. AWS Management Console. <https://us-west-2.console.aws.amazon.com/console>.
32. Stack overflow question: Do you get charged for a ‘stopped’ instance on EC2? <http://stackoverflow.com/questions/2549035/do-you-get-charged-for-a-stopped-instance-on-ec2>.
33. Бахмееев О.Ю. Deep learning software, 2016. <https://svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2015TheanoCuda/code>.
34. Попова М.С. Deep learning software, 2015. <https://svn.code.sf.net/p/mlalgorithms/code/Group174/Popova2015DeepLearning>.
35. Bishop C. M. Pattern recognition and machine learning. — New York, NY, USA: Springer-Verlag, 2006. 738 p.
36. Cassioli A., Lorenzo D. D., Locatelli M., Schoen F., Sciandrone M. Machine learning for global optimization // Comput. Optim. Appl., 2012. Vol. 1. No. 1. P. 279–303.

Поступила в редакцию 14.12.15

SYSTEMS AND MEANS OF DEEP LEARNING FOR CLASSIFICATION PROBLEMS

O. Yu. Bakhteev¹, M. S. Popova¹, and V. V. Strijov²

¹Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141700, Russian Federation

²A. A. Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The paper provides a guidance on deep learning net construction and optimization using graphics processing unit. The paper proposes to use GPU-instances on the cloud platform Amazon Web Services. The problem of time series classification is considered. The paper proposes to use a deep learning net, i. e., a multilevel superposition of models, belonging to the following classes: restricted Boltzman machines, autoencoders, and neural nets with softmax-function in output. The proposed method was tested on a dataset containing time segments from mobile phone accelerometer. The analysis of relation between classification error, dataset size, and superposition parameter amount has been conducted.

Keywords: time series classification; deep learning; model superposition; autoencoder; restricted Boltzmann machine; cloud service

DOI: 10.14357/08696527160201

Acknowledgments

This project is supported by the Russian Foundation for Basic Research (grant 16-37-00488).

References

1. Cho, K. 2014. Foundations and advances in deep learning. DSc. Espoo: Aalto University. 277 p.
2. Langkvist, M., L. Karlsson, and A. Loutfi. 2014. A review of unsupervised feature learning for time-series modelling. *Pattern Recogn. Lett.* 42(1):11–24.
3. Desell, T., S. Clachar, J. Higgins, and B. Wild. 2015. Evolving deep recurrent neural networks using ant colony optimization. *Evolutionary computation in combinatorial optimization*. Eds. G. Ochoa, and F. Chicano. Lecture notes in computer science ser. Springer. 9026:86–98.
4. Popova, M. S., and V. V. Strijov. 2015. Postroeniye neironnykh setey glubukogo obucheniya dlya klassifikatsii vremennykh ryadov [Building superposition of deep learning neural networks for solving the problem of time series classification]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 25(3):60–77.

5. Wager, S., S. Wang, and P. Liang. 2013. Dropout training as adaptive regularization. *Adv. Neur. In.* 26:351–359.
6. Wan, L., M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. 2013. Regularization of neural networks using dropconnect. *30th Conference (International) on Machine Learning Proceedings*. 1058–1066.
7. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15:1929–1958.
8. Gal, Y., and Z. Ghahramani. 2015. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. arXiv preprint arXiv:1506.02142. Available at: <http://arxiv.org/abs/1506.02142> (accessed November 25, 2015).
9. Goodfellow, I. J., Q. V. Le, A. M. Saxe, H. Lee, and A. Y. Ng. 2009. Measuring invariances in deep networks. *Adv. Neur. In.* 22:646–654.
10. Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. 2014. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199. Available at: <http://arxiv.org/abs/1312.6199> (accessed November 25, 2015).
11. Raiko, T., H. Valpola, and Y. LeCun. 2012. Deep learning made easier by linear transformations in perceptrons. *J. Mach. Learn. Res.* 22:924–932.
12. Bengio, Y., E. Laufer, G. Alain, and G. Yosinski. 2014. Deep generative stochastic networks trainable by backprop. *31st Conference (International) on Machine Learning Proceedings*. 226–234.
13. Ioffe, S., and C. Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167. Available at: <http://arxiv.org/abs/1502.03167> (accessed November 25, 2015).
14. Li, Z., C. Chang, F. Liang, T. S. Huang, C. Cao, and J. R. Smith. 2013. Learning locally-adaptive decision functions for person verification. *IEEE Conference on Computer Vision and Pattern Recognition Proceedings*. 3610–3617.
15. Sutskever, I., G. Hinton, and G. Taylor. 2009. The recurrent temporal restricted Boltzmann machine. *Adv. Neur. In.* 21:1601–1608.
16. Fischer, A., and C. Igel. 2014. Training restricted Boltzmann machines: An introduction. *Pattern Recogn.* 47:25–39.
17. Socher, R., E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Adv. Neur. In.* 24:801–809.
18. Shu, M., and F. Fyshe. 2013. Sparse autoencoders for word decoding from magnetoencephalography. *3rd NIPS Workshop on Machine Learning and Interpretation in NeuroImaging Proceedings*. Available at: <http://www.cs.cmu.edu/~afyshe/papers/SparseAE.pdf> (accessed November 25, 2015).
19. Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. 2013. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11:3371–3408.
20. Kwapisz, J. R., G. M. Weiss, and S. Moore. 2010. Activity recognition using cell phone accelerometers. *SIGKDD Explorations* 12(2):74–82.
21. Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. 2010. Theano: A CPU and GPU math

- expression compiler. *Conference on Python for Scientific Computing Proceedings*. 3–11.
22. Bastien, F., P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. 2012. Theano: New features and speed improvements. *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*. Available at: <http://arxiv.org/pdf/1211.5590v1.pdf> (accessed November 25, 2015).
 23. Goodfellow, I. J., D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. 2013. PyLearn2: A machine learning research library. arXiv preprint arXiv:1308.4214. Available at: <http://arxiv.org/abs/1308.4214> (accessed November 25, 2015).
 24. Dieleman, S., J. Schluter, C. Raffel, et al. 2015. Lasagne: First release. August 13, 2015. Available at: <http://dx.doi.org/10.5281/zenodo.27878> (accessed November 25, 2015).
 25. Nickolls, J., I. Buck, M. Garland, and K. Skadron. 2008. Scalable parallel programming with CUDA. *ACM Queue* 6(2):40–53.
 26. Stone, J. E., D. Gohara, and G. Shi. 2010. OpenCL: A parallel programming standard for heterogeneous computing systems. *J. IEEE Design Test* 12(10):66–73.
 27. Erhan, D., Y. Bengio, A. Courville, P. Manzagol, and P. Vincent. 2010. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* 11:625–660.
 28. Duan, K., S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo. 2003. Multi-category classification by soft-max combination of binary classifiers. *4th Workshop (International) on Multiple Classifier Systems*. 125–134.
 29. Cho, K., T. Raiko, and A. Ilin. 2013. Gaussian–Bernoulli deep Boltzmann machine. *2013 Joint Conference (International) on Neural Networks*. 1–7.
 30. Hinton, G. E., S. Osindero, and Y. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Comput.* 18:1527–1554.
 31. AWS Management Console. Available at: <https://us-west-2.console.aws.amazon.com/console/> (accessed November 25, 2015).
 32. Stack overflow question: Do you get charged for a ‘stopped’ instance on EC2? Available at: <http://stackoverflow.com/questions/2549035/do-you-get-charged-for-a-stopped-instance-on-ec2> (accessed November 25, 2015).
 33. Bakhteev, O. Yu. 2016. Deep learning software. Available at: <https://svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2015TheanoCuda/code/> (accessed November 25, 2015).
 34. Popova, M. S. 2015. Deep learning software. Available at: <https://svn.code.sf.net/p/mlalgorithms/code/Group174/Popova2015DeepLearning/> (accessed November 25, 2015).
 35. Bishop, C. M. 2006. *Pattern recognition and machine learning*. New York, NY: Springer-Verlag. 738 p.
 36. Cassioli, A., D. D. Lorenzo, M. Locatelli, F. Schoen and Sciandrone, M. 2012. Machine learning for global optimization. *Comput. Optim. Appl.* 1(1):279–303.

Received December 14, 2015

Contributors

Bakhteev Oleg Y. (b. 1993)— student, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141700, Russian Federation; bakhteev@phystech.edu

Popova Maria S. (b. 1994)— student, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141700, Russian Federation; maria.popova@phystech.edu

Strijov Vadim V. (b. 1967) — Doctor of Science in physics and mathematics, leading scientist, A. A. Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation; strijov@ccas.ru

АНАЛИЗ САМОСИНХРОННОСТИ ЭЛЕКТРОННЫХ СХЕМ НА НИЖНЕМ УРОВНЕ ИЕРАРХИИ*

Л. П. Плеханов¹

Аннотация: Самосинхронные электронные схемы, обладающие уникальными свойствами, требуют обязательного анализа на свойство самосинхронности. При анализе необходимо вычислить и проверить все рабочие состояния схемы и переходы между ними. Существующие событийные методы оперируют всеми уравнениями схемы одновременно. Для практических схем большого размера анализ этими методами приводит к неприемлемым вычислительным затратам. Предложенный ранее автором функциональный иерархический метод позволяет анализировать схемы по частям «снизу вверх». Уравнения схемы анализируются только на нижнем уровне иерархии, на верхних уровнях используются взаимосвязи фрагментов и параметры, полученные на нижних уровнях. Метод обеспечивает эффективный анализ схем всех возрастающих размеров. Подробно описан функциональный метод на нижнем уровне иерархии.

Ключевые слова: самосинхронные схемы; асинхронные схемы; анализ самосинхронности; иерархический анализ

DOI: 10.14357/08696527160202

1 Введение

Самосинхронные схемы (СС-схемы) обладают уникальными свойствами, не достижимыми в реализации других типов схем, синхронных или асинхронных. К ним относятся независимость поведения от задержек элементов, полное отсутствие состязаний, отказобезопасность, правильность функционирования в максимально широком диапазоне внешних условий (температуры и напряжения питания) и некоторые другие. Эти свойства были обоснованы теоретически и проверены экспериментально [1–3].

Одна из главных проблем проектирования СС-схем состоит в необходимости анализа самосинхронности практических схем, т. е. схем любого размера. Классический подход, основанный на теории полумодулярности Д. Е. Маллера [4] и развитый группой В. И. Варшавского [5], позволяет анализировать поведение схемы (переключение элементов) из одного начального состояния. Для практики этого явно недостаточно, так как необходимо обеспечить полумодулярность для

* Исследование выполнено при частичной финансовой поддержке по Программам фундаментальных исследований 2016 г. Президиума РАН (проект 0063-2015-0015 РАН 1.33П) и подпрограмме № 4 ОНИТ РАН на 2016 г. (проект 0063-2015-0016 III.3).

¹ Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, lplekhanov@inbox.ru

всех возможных начальных состояний и входных воздействий, что приводит к проблеме полноты анализа [6]. Проблема полноты в классическом подходе даже не упоминалась.

Трудности создания практических СС-схем хорошо освещены в [1], они же справедливы и в наше время. За рамками обсуждений, однако, осталась проблема вычислительной сложности при анализе получаемых решений. Эта проблема и стала главным препятствием на пути проектирования больших схем — состоящих из десятков и более элементов.

Классический анализ схем на свойство полумодулярности основан на изучении состояний схемы и переключений элементов. Состояние схемы — это множество значений (0 или 1) выходов всех элементов схемы. Перед тем как переключиться, элемент переходит в возбужденное состояние, при котором в нем происходят внутренние переходные процессы. Эти процессы либо могут закончиться естественно — изменением выхода, либо могут быть прерваны без изменения выхода из-за изменений входов. Последняя ситуация называется конфликтом и является потенциальным источником состязаний.

Первый метод анализа, предложенный Маллером [4], — метод диаграмм переходов (ДП). Цель анализа — выявить конфликты в работе схемы. Схемы без конфликтов называются полумодулярными.

Вычислительная проблема в методе ДП возникает в связи с тем, что при изменении состояния могут возбудиться несколько элементов, что характеризует параллельность процессов в схеме. Пусть N — число возбужденных элементов в состоянии (степень параллельности). По правилам построения ДП данное состояние порождает N следующих состояний, в каждом из которых $N - 1$ возбужденных элементов. Каждое из этих состояний, в свою очередь, порождает $N - 1$ последователей, в которых по $N - 2$ возбуждений. Налицо факториальная зависимость вычислений от степени параллельности.

Повышение параллельности вычислений — магистральный путь развития электроники, и метод ДП становится неэффективным. Например, если встретится состояние с $N = 10$, что вполне реально, то число порожденных им состояний превысит 3,6 млн. Такие состояния могут возникать в разных частях ДП, последовательно и параллельно.

Группой В. И. Варшавского был разработан метод диаграмм изменений (ДИ) [5]. Метод позволяет более компактно представить работу схемы, когда описываются не полные состояния, а только изменяющиеся выходы элементов. Метод основан на анализе эквивалентной ДП. Поскольку все проходимые состояния схемы должны быть проанализированы при имеющейся степени параллельности, вычислительные трудности хотя и уменьшаются, но остаются большими. Они переходят на другие аспекты анализа, например, на переход от описания схемы, заданной в элементах, к ДИ, и особенно на обработку процессов, порожденных элементами с операцией «ИЛИ».

Подход, реализуемый обоими приведенными классическими методами, будем называть далее событийным, так как его методы описывают и анализируют

события — переключение элементов схемы. С этой точки зрения ДП и ДИ — разные способы представления событий.

Общее для событийного подхода то, что схема должна быть представлена в замкнутом виде, гарантирующем ее самогенерацию, а также то, что анализ производится начиная только с одного заданного начального состояния. Оба эти обстоятельства требуют искусственных построений и затрудняют проектирование схем.

Для практических целей необходимо проанализировать все состояния, которые проходит схема в реальной работе, т. е. обеспечить полноту анализа.

Для обеспечения полноты в событийном подходе можно провести множество сеансов анализа по числу возможных начальных состояний. Помимо того, что это множество экспоненциально зависит от числа элементов схемы, такая стратегия не рациональна, так как не все возможные состояния реализуются при работе схемы.

Источником изменений в схеме являются изменения ее входов. Для анализа необходимо подобрать такую последовательность входных значений, чтобы пройти в одном сеансе все рабочие состояния схемы независимо от начального.

Проведенные в [6] расчеты показывают, что в событийном подходе гарантировать выполнение этого условия можно, подавая на вход 2^{2I+M} наборов, где I — число информационных входов, M — число переменных памяти, и эти наборы должны следовать в определенном порядке. Это означает, что необходимо строить весьма сложное замыкание, увеличивая число уравнений для анализа. Подобного или другого способа решения проблемы полноты в событийном подходе до сих пор не предложено.

Изложенные вычислительные трудности ограничивают применение событийных методов: для ДП — схемами из двух–трех десятков элементов, для ДИ — несколько больше (практического опыта с полнотой пока нет).

Методы событийного подхода являются универсальными, они анализируют любую предъявленную схему, без учета ее особенностей. Как известно, универсальные методы наиболее затратны, и решение проблемы, как правило, следует искать в специализированных методах.

Кроме того, событийные методы анализируют схему целиком. На практике конечную схему обычно уже невозможно анализировать целиком из-за ее размера, что не гарантирует свойства полумодулярности.

Единственным путем анализа схем любого размера представляется применение иерархического метода, гарантирующего самосинхронность составной схемы на основании ранее выполненного анализа ее частей. В событийном подходе такого метода также не предложено.

2 Самосинхронные схемы и иерархический метод

Определение «самосинхронные схемы» было введено в книге [1, с. 73] (в оригинале — *aperiodические* или *самосинхронизирующиеся*, но эти термины не

прижились) как схемы, правильность функционирования которых не зависит от величин задержек элементов. Из теории Маллера [4] следует, что если схема замкнута и полумодулярна начиная с некоторого начального состояния, то она самосинхронна (но только с этим начальным состоянием). Традиционно самосинхронность проверялась именно замыканием и анализом полумодулярности.

Однако для практического проектирования такой способ проверки всей схемы, как указывалось выше, неприемлем из-за вычислительных трудностей. Иерархический метод подразумевает разработку отдельных самосинхронных частей (далее — СС-фрагментов) и объединение их в общую СС-схему.

В классическом подходе при таком способе приходится замыкать и размыкать каждый СС-фрагмент. Теоретически замыкание — имитация поведения схемы следующего верхнего уровня — должно обеспечивать необходимую полноту переключений входов схемы. В классическом подходе эта проблема пока не решена. На практике, как показывает опыт, из-за сложности полного замыкания его никогда не делают, кроме тривиальных случаев.

Упомянутые проблемы проектирования СС-схем могут быть решены с помощью предложенного ранее функционального подхода [7, 8].

Особенности функционального подхода, по сравнению с событийным, следующие:

- анализируются уравнения элементов, состояния схемы не вычисляются;
- рассматриваются разомкнутые схемы, без замыканий;
- учитываются свойства СС-схем [1]: самосинхронное кодирование данных, двухфазный порядок работы, индицирование сигналов и др.

В функциональном подходе принято также другое определение СС-схемы, более удобное для проектирования. Это определение удовлетворяет определению в [1] и конкретизирует его [8].

Самосинхронная схема — это разомкнутая схема, во всех реальных начальных и проходимых при работе состояниях имеющая два свойства: отсутствие состязаний при любых конечных задержках элементов и отказобезопасность по отношению к константным залипаниям на 0 и 1 выходов элементов.

Под отказобезопасностью [9] здесь понимается остановка работы схемы при возникновении указанных неисправностей.

Отметим, что при этом подходе СС-фрагменты не нуждаются в замыкании—размыкании для анализа, а только схема целиком в конечном применении должна быть замкнута для автоматического чередования фаз — самогенерации.

В иерархическом методе в рамках функционального подхода только СС-фрагменты нижнего уровня представляются и анализируются в форме булевых уравнений. На верхних уровнях анализируются взаимосвязи фрагментов. В результате вычислительная сложность этого метода практически линейна по числу сигналов и фрагментов, что позволяет анализировать «поэтажно», снизу вверх, схемы любого размера. В публикациях [7, 8] анализ на нижнем уровне приводится в достаточно общем виде.

Темой данной статьи является более подробное изложение анализа СС-схем на нижнем уровне иерархии.

3 Постановка задачи

Особенности СС-схем заключаются в специальном — самосинхронном — кодировании информации и двухфазном режиме работы [1]. Самосинхронная схема автоматически переходит поочередно из одной фазы в другую, что требуется для обеспечения самосинхронности. Фазы носят название *рабочей* и *спейсера* (промежуточной).

Исходными данными для анализа послужит разомкнутая схема, представленная в булевых уравнениях элементов — зависимости выходной переменной элемента от входных переменных. (В соответствии с теорией Маллера одному уравнению соответствует один элемент схемы.)

Также необходимо явно обозначить интерфейс схемы — все внешние входы и выходы с их типами, специфичными для самосинхронности (*СС-типами*).

Введем уточнения и определения для дальнейшего изложения.

БСЯ — бистабильная ячейка — известная в электронике пара элементов с перекрестными связями, имеющая два стабильных состояния.

Самосинхронные типы сигналов могут быть следующими:

У-сигналы — унарные входные управляющие сигналы;

И-сигналы — унарные выходные индицирующие сигналы;

ПФС-сигналы — парафазные сигналы со спейсером, информационные — два сигнала в парафазном коде;

Фазовые сигналы — сигналы, значения которых определяются текущей фазой работы, это *У-*, *И-* и *ПФС-сигналы*;

БИН-сигналы — бинарные информационные сигналы — два связанных сигнала (обычно это выходы бистабильной ячейки), нефазовые;

УИН-сигналы — унарные информационные нефазовые сигналы (например, каждый из БИН-сигналов, рассматриваемый независимо).

Другие разновидности сигналов (режимные, установочные и др.) могут быть легко учтены и для простоты рассматриваться не будут.

В интерфейсе схемы должны присутствовать хотя бы по одному фазовому сигналу на входе и выходе. Также должны быть указаны значения спейсеров, 0 или 1, для входных фазовых сигналов.

Фаза — процесс, начинающийся с момента перехода входных фазовых сигналов в значения данной фазы и заканчивающийся установлением всех выходных фазовых сигналов в значения, соответствующие этой же фазе.

Цикл — процесс, состоящий из двух последовательных фаз. Первая фаза в цикле называется *предыдущей*, вторая (основная для анализа) — *основной*. Всего

существует два варианта цикла: начинающийся со спейсера и начинающийся с рабочей фазы.

Следуя [1], будем отождествлять имя элемента с именем его выходного сигнала.

Дисциплиной входных сигналов далее будет называться порядок их изменений относительно друг друга.

Задача ставится таким образом:

- установить самосинхронность схемы в соответствии с приведенным выше определением;
- получить информацию для включения в интерфейс для анализа на следующем верхнем уровне иерархии.

4 Метод анализа

Метод основан на проверке двух свойств предъявленной для анализа схемы в соответствии с приведенным выше определением СС-схемы.

Как показано в [1], свойство отказобезопасности обеспечивается другим свойством СС-схем, более удобным для анализа, — *индцируемостью*.

Смысль индицируемости (строгое определение дано в [1]) в том, что любое изменение выхода каждого элемента схемы должно «отслеживаться» ее выходными фазовыми сигналами. Если все изменения выходов произошли, то и все фазовые выходы схемы должны перейти к значениям текущей фазы. Если хотя бы один выход элемента «залип» на константу 0 или 1, то хотя бы один из фазовых выходов схемы остался в значении предыдущей фазы.

Индицируемость определена для разомкнутых схем, поскольку в замкнутых схемах входов и выходов нет и понятие фазы неприменимо.

Одно из основных теоретических положений для разомкнутых схем [1] утверждает: «*Двухфазная комбинационная схема является апериодической тогда и только тогда, когда она индицируема*».

Здесь апериодичность подразумевает отсутствие состязаний.

Отметим свойства сигналов в СС-схемах, следующие из теории [1].

Любой сигнал в каждой из фаз может меняться не более 1 раза.

Фазовый сигнал, если меняется, то в заданной фазе всегда в одном направлении, либо из 0 в 1, либо из 1 в 0. Эта особенность, в частности, и позволила доказать упомянутое утверждение.

Иная ситуация с БИН (УИН) сигналами. Эти сигналы, в отличие от фазовых, в заданной фазе могут меняться в разных направлениях, и такая особенность может порождать состязания.

Таким образом, если схема не комбинационна, то ее индицируемость необходима, но не достаточна. Для самосинхронности требуется проверка состязаний.

Идея анализа состоит в том, чтобы выделить в схеме источники БИН- и УИН-сигналов — БСЯ — и проанализировать состязания, которые могут

ими порождаются. Проверка состязаний может проводиться для каждой БСЯ отдельно, что не требует значительных вычислительных ресурсов. Оставшаяся часть схемы, если она индицируема, состязаний не имеет.

С целью обеспечения полноты для всех входных парных сигналов схемы (ПФС и БИН) вводятся независимые величины — *параметры анализа* [8].

Здесь и далее в булевых выражениях знак \wedge обозначает отрицание, знак \vee — операцию ИЛИ, отсутствие знака — операцию И.

Введем обозначения: A_i и B_i — сигналы i -й ПФС-пары; C_j и D_j — сигналы j -й БИН-пары; символы @ и # будут признаками параметров.

Анализ проводится для каждого из двух циклов.

Сначала делается расчет установившихся значений всех элементов схемы в предыдущей фазе. Начальные значения внутренних сигналов в этой фазе принимаются произвольными. На ПФС-входы, если фаза — спейсер, подаются значения $A_i = B_i = S_i$, где S_i — константа 0 или 1 — значение соответствующего спейсера. Если фаза рабочая, значения пары будут инверсны: $A_i = @A_i$; $B_i = ^{@A_i}$, где $@A_i$ — первый параметр пары. На БИН-входы всегда подаются инверсные значения: $C_j = @C_j$; $D_j = ^{@C_j}$. Значения У-выходов должны соответствовать фазе.

Далее проводится расчет значений в основной фазе. Внутренние начальные значения для нее остаются теми, что получились в предыдущей фазе. В отличие от предыдущей, на парные входы подаются другие параметры. На ПФС-входы в рабочей фазе подаются значения $A_i = #A_i$ и $B_i = ^{#A_i}$, на БИН-входы — $C_j = #C_j$ и $D_j = ^{#C_j}$.

В каждом цикле после расчета в предыдущей фазе сигналы, оставшиеся произвольными, принадлежат БСЯ, так как хранят свое значение и служат признаком для выявления этих БСЯ.

Все БСЯ схемы определяются после расчетов обоих циклов. Выявленные значения произвольных сигналов (с признаком \$) добавляются в параметры анализа.

Введенные параметры анализа и расчеты в двух циклах обеспечивают необходимую полноту анализа как по значениям и изменениям входных сигналов, так и по начальным состояниям.

Полученные результаты расчетов в каждом цикле используются как основа для дальнейшего анализа. Информация для верхнего уровня иерархии вычисляется в процессе анализа.

4.1 Проверка индицируемости сигналов

Индицируемость внутренних сигналов схемы обязательна для самосинхронности. Внешние сигналы могут либо индицироваться, либо нет на фазовых выходах схемы, и эта информация входит в информацию для верхнего уровня иерархии.

Индцируемость некоторого сигнала A в [1] определена как булева производная функции выходного элемента схемы по A с учетом транслируемости — передачи изменений по цепочке элементов от A до выхода. Такое определение позволяет установить самосинхронность только комбинационных схем, что и отражено в цитированном выше утверждении. Для схем с памятью транслируемость не соблюдается, и данное определение не помогает анализу.

В функциональном подходе индцируемость как одно из требований определения самосинхронности должна применяться к схемам общего вида. Поэтому в данном подходе она определена конструктивно — имитацией залипания — и проверяется прямым методом.

Делается новый расчет основной фазы таким образом, как если бы анализируемый сигнал залип — не менялся в этой фазе. Если все фазовые выходы схемы перешли в состояние, соответствующее основной фазе (как в первоначальном расчете), залипание не замечено и сигнал не индцируется. Параметры анализа при этом дадут информацию о сигналах и условиях этого нарушения, что позволит эффективно исправить схему. (Для сравнения — в событийных методах, чтобы установить место в схеме, порождающее нарушение полумодулярности, требуются дополнительные вычисления.)

Проверка индцируемости с параметрами анализа более подробно изложена в [8] и здесь повторяться не будет.

4.2 Проверка состязаний

Как указывалось выше, источником возможного состязания в СС-схемах является УИН-сигнал, а само состязание может возникнуть на элементе, на который этот сигнал подается.

Введем термины:

плечо — один из элементов БСЯ;

источник — источник УИН-сигнала;

последователь — элемент, на вход которого подается УИН-сигнал от источника.

Источниками в схеме будут плечи БСЯ, входные БИН- и УИН-сигналы, а также сигналы, идущие от источников через инверторы или повторители. Последователями могут быть как плечи других БСЯ, так и элементы, не входящие в БСЯ, например индикаторы.

Состязания будут анализироваться по взаимодействию источник–последователь. Главную роль в проверке будет играть дисциплина их изменений.

Введем понятие блокирования.

Элемент будет считаться *заблокированным*, если значения его фазовых входов не позволяют изменяться выходному сигналу.

На рис. 1 показана диаграмма корректного взаимодействия источника и последователя. Здесь Q и Y — сигналы соответственно источника и последователя;

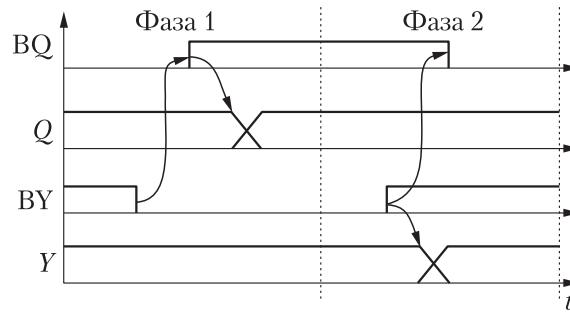


Рис. 1 Взаимодействие источника и последователя

BQ и BY — управляющие сигналы источника и последователя, значения этих сигналов: 0 — заблокировано, 1 — разрешено.

Блокирующая фаза (блок-фаза) — фаза, в которой элемент переходит из неблокированного состояния в заблокированное.

На рис. 1 блок-фаза для источника — фаза 2, для последователя — фаза 1.

Как следует из рис. 1, принципиально важное условие корректности заключается в том, что сигнал BQ не должен меняться раньше сигнала BY. В противном случае, в условиях произвольности задержек в СС-схемах, выход источника Q и сигнал BY могут состязаться, что вызовет непредсказуемые изменения выхода последователя Y .

Вследствие необходимой корректности сигналы BQ и BY должны быть связаны, а именно: это может быть один и тот же сигнал либо BQ должен всегда изменяться позже BY, т. е. BQ должен быть соединен с BY через промежуточные элементы.

Условие корректности, таким образом, может быть определено по соединениям элементов схемы. Полезными здесь оказываются блокирующие функции.

Блокирующей функцией (блок-функцией) элемента назовем булеву производную его функции по всем входным УИН-сигналам.

Если блок-функция равна 0, элемент заблокирован, если 1, то выход элемента может меняться.

Блокирующие сигналы (блок-сигналы) — это фазовые сигналы, входящие в блок-функцию.

В СС-схемах обычно блок-функции в представлении ДНФ имеют вид одной импликанты и блок-сигналы входят в нее как сомножители. Иные варианты на практике не используются из-за того, что при этом не удается организовать правильное взаимодействие и индикацию. Поэтому другие варианты далее не рассматриваются.

Отсутствие состязаний последователя выявляется двумя расчетами: проверкой соединений блок-сигналов и проверкой монотонности переходов элементов.

4.2.1 Проверка соединений блокирующих сигналов

Назовем сигнал A инициатором элемента U , если вход этого элемента соединен с A через цепочку промежуточных элементов.

Пусть вычислены блок-функции источника и последователя. Тогда условие корректности на рис. 1 можно сформулировать так:

Хотя бы один блок-сигнал последователя должен быть либо блок-сигналом источника, либо инициатором блок-сигнала источника.

Это необходимое условие должно выполняться для всех пар источник–последователь в схеме и проверяется по соединениям элементов.

4.2.2 Проверка монотонности переходов последователя

Последователь может иметь на входе один или несколько УИН-сигналов. Если УИН-сигнал один, он не состязается со своими блок-сигналами по предыдущему условию.

Если последователь — плечо БСЯ, то у него на входе всегда два УИН-сигнала, один из них — от второго плеча. Сигнал от второго плеча, если изменяется, то всегда в том же направлении, что и сигнал первого плеча, так как оба они связаны друг с другом через два инвертирующих (или неинвертирующих) элемента. По этой причине состязаний здесь также не возникает.

Поскольку УИН-сигналы в одной фазе могут изменяться в разных направлениях, требуется проверять монотонность переходов последователя.

Согласно [1] функция $f(h_1, \dots, h_n)$ называется изотонной по переменной h_i , если $f(h_i = 1) \geq f(h_i = 0)$, и антитонной, если $f(h_i = 1) \leq f(h_i = 0)$.

Под монотонностью переходов понимается изменение всех входов элемента так, чтобы выполнялась либо изотонность по всем входам, либо антитонность по всем входам. Неменяющиеся входы не учитываются.

Если монотонность реализуется, состязаний не возникает.

Таким образом, проверка на монотонность подлежат последователи, не являющиеся плечами БСЯ и имеющие на входе более одного УИН-сигнала. Проверка осуществляется в основной фазе цикла. Если на входе последователя есть блок-сигналы, то проверка делается только в неблокирующей основной фазе, если блок-сигналов нет, то в основных фазах обоих циклов.

Для проверки монотонности введем вспомогательные *изопеременные* — переменные, всегда изменяющиеся из 0 в 1.

В результате предварительного расчета для последователя получены выражения его входов через параметры анализа в обеих фазах цикла.

Введем обозначения:

P — множество параметров анализа;

$A(P)$ — выражение выхода последователя в предыдущей фазе;

$Z(P)$ — то же в основной фазе;

$U_i(P)$ — выражение i -го сигнала в предыдущей фазе;

$V_i(P)$ — то же в основной фазе;

h_i — изопеременная;

H — множество изопеременных;

$G_i = U_i \wedge h_i \vee V_i h_i$ — вспомогательная функция сигнала.

Вспомогательная функция по построению меняется от выражения сигнала в предыдущей фазе к выражению в основной фазе, когда изопеременная меняется из 0 в 1.

В этих обозначениях выход последователя переходит из 0 в 1 при условии $\wedge A(P)Z(P) = 1$, а из 1 в 0 — при условии $A(P) \wedge Z(P) = 1$.

В функцию последователя подставим вместо реальных переменных их вспомогательные функции. В результате получится выражение выхода последователя через параметры анализа и изопеременные: $F(P, H)$.

Рассмотрим теперь две проверочные функции:

$$F_1 = \wedge A(P)Z(P)F(P, H);$$

$$F_2 = A(P) \wedge Z(P)F(P, H).$$

Состязания будут отсутствовать при выполнении двух условий:

- (1) функция F_1 должна либо равняться нулю, либо быть изотонной по всем изопеременным;
- (2) функция F_2 должна либо равняться нулю, либо быть антитонной по всем изопеременным.

Если условия нарушены, то те параметры анализа и изопеременные, при которых нарушение произошло, покажут детальную диагностику этого нарушения.

Монотонность функции $f(h_1, \dots, h_n)$ по переменной h_i проверяется по условиям:

- для изотонности: $f(h_i = 0) \wedge f(h_i = 1) = 0$;
- для антитонности: $\wedge f(h_i = 0) f(h_i = 1) = 0$.

Пример 1. Рассмотрим схему несамосинхронного двухступенчатого триггера (рис. 2).

Элементы триггера описываются уравнениями:

$$U_1 = \wedge(D_1 C \vee U_2);$$

$$U_2 = \wedge(D_2 C \vee U_1);$$

$$C_Z = C;$$

$$Y_1 = \wedge((U_1 \vee C_Z) Y_2);$$

$$Y_2 = \wedge(Y_1(C_Z \vee U_2));$$

$$I = \wedge(U_1 Y_1 \vee D_1 U_2 C_Z \vee C_Z U_1 D_2 \vee U_2 Y_2).$$

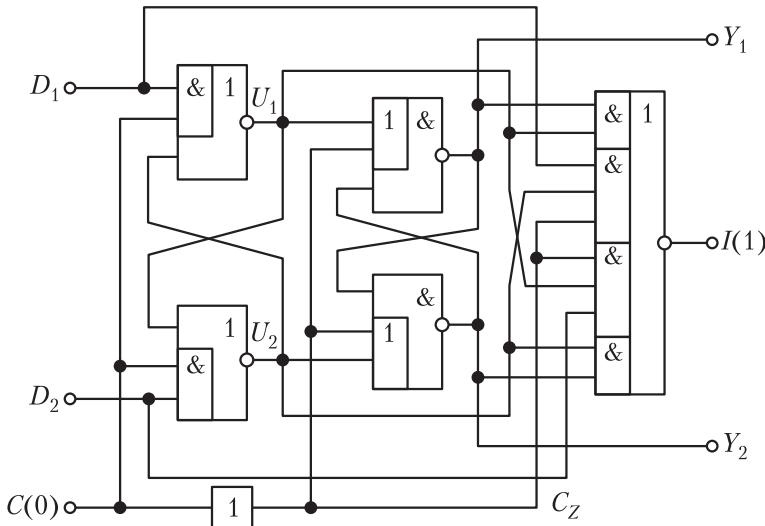


Рис. 2 Триггер с повторителем в цепи Y -сигнала: D_1 и D_2 — входные БИН-сигналы; Y_1 и Y_2 — выходные БИН-сигналы; C — Y -сигнал; I — И-сигнал; в скобках показаны значения спейсеров

Схема не самосинхронна из-за наличия повторителя в цепи Y -сигнала.

Блок-функцией для обоих плеч левой БСЯ будет C ; для плеч правой БСЯ — $\wedge C_Z$; для индикаторного элемента — C_Z .

Рассмотрим пару: элементы U_1 и Y_1 . На рис. 2 видно, что блок-сигнал C_Z последователя Y_1 не является инициатором блок-сигнала C плеча U_1 . Такова же ситуация относительно пар U_1 и I , U_2 и Y_2 , U_2 и I .

Но это не единственное нарушение самосинхронности в схеме. Анализ индикации показывает, что в фазе спейсера сигнал C_Z не индицируется.

Условием индикации этого сигнала через параметры анализа будет

$$\wedge @D_1 \#D_1 \$U_1 \vee @D_1 \wedge \#D_1 \wedge \$U_1 = 0,$$

где $@D_1$ и $\#D_1$ — параметры предыдущей (рабочей) и основной фаз соответственно; $\$U_1$ — запомненное значение сигнала U_1 в предыдущей фазе.

Нарушение индикации реализуется, когда одна из импликант условия обратится в 1. По первой импликанте нарушение будет, если сигнал U_1 в предыдущей фазе запомнил 1, сигнал D_1 переходит из 0 в 1, а сигнал D_2 — из 1 в 0. (При анализе учитывается, что внешняя схема обеспечивает дисциплину входов: изменения сигналов D_1 и D_2 происходят после изменения сигнала C .)

Пример 2. На рис. 3 показана измененная схема предыдущего примера, также несамосинхронная.

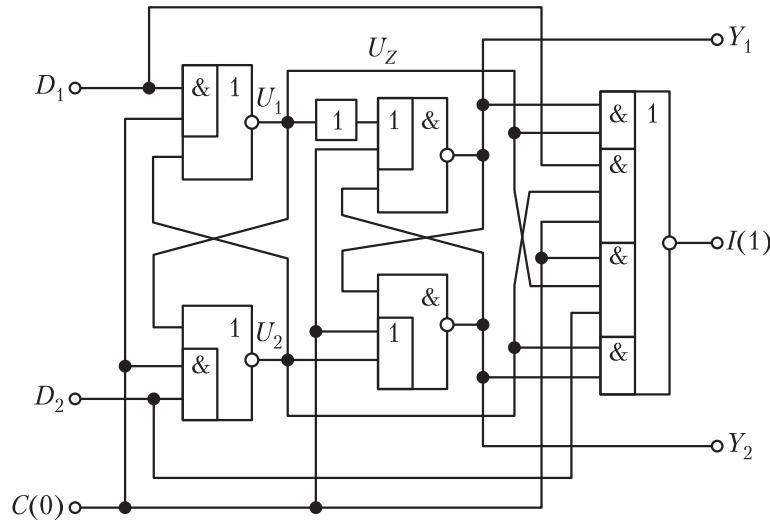


Рис. 3 Двухступенчатый триггер с повторителем в цепи УИН-сигнала: D_1 и D_2 — входные БИН-сигналы; Y_1 и Y_2 — выходные БИН-сигналы; C — У-сигнал; I — И-сигнал; в скобках даны значения спейсеров

Схема описывается уравнениями:

$$\begin{aligned} U_1 &= \wedge(D_1 C \vee U_2); \\ U_2 &= \wedge(U_1 \vee CD_2); \\ U_Z &= U_1; \\ Y_1 &= \wedge((U_Z \vee C)Y_2); \\ Y_2 &= \wedge(Y_1(C \vee U_2)); \\ I &= \wedge(Y_1 U_1 \vee D_1 U_2 C \vee C U_1 D_2 \vee U_2 Y_2). \end{aligned}$$

В этой схеме выполняются условия корректности соединений блокирующих сигналов. Но проверка монотонности элемента I выявляет нарушение в рабочей фазе.

Вспомогательные функции сигналов в этой (основной) фазе имеют вид (изо-переменные обозначены признаком %):

$$\begin{aligned} C &= \%C; \\ D_1 &= @D_1 \wedge \%D_1 \vee \#D_1 \%D_1; \\ D_2 &= \wedge @D_1 \wedge \%D_2 \vee \wedge \#D_1 \%D_2; \\ U_1 &= \wedge @D_1 \wedge \%U_1 \vee \wedge \#D_1 \%U_1; \end{aligned}$$

$$\begin{aligned}U_2 &= @D_1 \wedge \%U_2 \vee \#D_1 \%U_2 ; \\Y_1 &= @D_1 ; \\Y_2 &= \wedge @D_1 ,\end{aligned}$$

где $@D_1$ и $\#D_1$ — параметры предыдущей (спейсера) и основной фаз соответственно.

Подстановка вспомогательных функций в функцию элемента I дает:

$$\begin{aligned}F = @D_1 \#D_1 \wedge \%C \vee \wedge @D_1 \wedge \%C \wedge \%U_1 \vee \wedge @D_1 \wedge \#D_1 \wedge \%C \vee \\ \vee \wedge @D_1 \wedge \%C \wedge \%U_2 \vee @D_1 \wedge \#D_1 \wedge \%U_1 \%U_2 \vee \wedge @D_1 \#D_1 \%U_1 \wedge \%U_2 .\end{aligned}$$

В рабочей фазе сигнал I переходит из 1 в 0, поэтому его первая проверочная функция равна нулю, вторая совпадает с F , и эта функция должна быть антитонной по всем изопеременным.

Проверка антитонности показывает, что по переменной $\%U_1$ функция не антитонна и нарушение будет при условии

$$\wedge @D_1 \#D_1 \%C \wedge \%U_2 = 1 .$$

Это условие означает, что состязания возможны при переходе сигнала C из 0 в 1, D_1 из 0 в 1, D_2 из 1 в 0, а сигнал U_2 еще не успел измениться (так как по условию должно быть $\%U_2 = 0$).

Аналогично функция F не антитонна и по переменной $\%U_2$ при условии

$$@D_1 \wedge \#D_1 \%C \wedge \%U_1 = 1 .$$

4.3 Оценка сложности вычислений на нижнем уровне

Пусть N — число элементов схемы; M — число ее информационных входов; P — число элементов памяти (триггеров). Оцениваться будет зависимость вычислений от этих параметров.

Булевы функции представим в алгебраической нормальной форме (АНФ — многочлен Жегалкина) — сумм по mod-2 конъюнктивных термов аргументов. Если число аргументов функции равно n , ее АНФ содержит не более $2^n - 1$ термов. Операций для двух АНФ может быть две: умножение и сумма по mod-2. Умножение двух АНФ состоит в умножении каждого терма первой на каждый терм второй (всего не более 2^{2n} умножений), затем в сравнении полученного терма с остальными (всего не более 2^n сравнений) и добавлении/удалении терма. Операции над термами будем считать элементарными, для практических схем несущественно зависящими от n .

Предварительный расчет значений в двух циклах проводится с функциями от не более чем $M + P$ аргументов, и затраты на него пропорциональны $4N$.

Затраты на дополнительный расчет с залипанием для каждого элемента пропорциональны $2N$, для всех элементов — $2N^2$. Суммарно эти затраты зависят от N полиномиально с порядком 2, а от остальных параметров — как $2^{2(M+P)}$.

Вычисления на состязания проводятся по парам. С учетом того, что для одного источника может быть несколько последователей, число пар будет от $N/2$ до $N - 1$. Определение соединений пар пропорционально числу пар.

Число аргументов проверочных функций при вычислении монотонности не более $M + P + L$, где L — число входов проверяемого элемента. Обычно L может достигать 7–8, но не увеличивается с ростом M или N .

В итоге можно заключить, что вычислительные затраты зависят от N полиномиально с порядком 2, а от остальных параметров — как $2^{2(M+P)}$.

4.4 Информация для верхнего уровня

Для схемы на верхнем уровне иерархии также должны выполняться условия самосинхронности, т. е. обеспечиваться индикация ее сигналов и взаимодействие источников и последователей по механизму, показанному на рис. 1.

Пусть на входе схемы есть входной УИН-сигнал R . Внутри схемы этот сигнал может иметь несколько последователей. В качестве общей блок-функции для R необходимо брать булеву сумму блок-функций всех его последователей. Блок-сигналами для R будут сигналы из общей функции.

На верхнем уровне сигнал R будет играть роль последователя, поэтому для выполнения дисциплины необходимо знать, какими фазовыми входами схемы этот сигнал блокируется и есть ли задержки в цепях блокирования. Для каждого блок-сигнала R по соединениям элементов следует определить, является ли он фазовым входом схемы или он соединен с фазовыми входами через промежуточные элементы.

Если хотя бы для одного блок-сигнала R не найдено соединение с фазовыми входами схемы, на верхнем уровне невозможно соблюсти требуемую дисциплину и анализируемая схема признается несамосинхронной.

Если все блок-сигналы R являются также входами схемы, для верхнего уровня достаточно привести их список.

Рассмотрим случай, когда внутренний блок-сигнал A соединен с фазовым входом схемы U через цепочку элементов. Для соблюдения дисциплины на верхнем уровне сигнал A должен быть инициатором сигнала U . Единственной возможностью для этого будет соединение сигнала A через цепочки элементов с фазовыми выходами схемы и затем через внешнюю обратную связь соединение с сигналом U . Для верхнего уровня в этом случае следует привести сигнал U и фазовые выходы, соединенные с сигналом A .

Выходные БИН- и УИН-сигналы на верхнем уровне будут играть роль источников, и задержка здесь не имеет значения.

Таким образом, информация, подаваемая наверх о схеме нижнего уровня, должна содержать следующее:

- (1) для выходных фазовых сигналов — список индицируемых ими входных и выходных сигналов схемы в каждой фазе;
- (2) для каждого входного БИН- и УИН-сигнала — блок-фазу и входные блок-сигналы. Если входной блок-сигнал имеет задержку в цепи блокирования, то указываются соответствующие фазовые выходы схемы;
- (3) для каждого выходного БИН- и УИН-сигнала — блок-фазу и входные блок-сигналы.

Пример 3. Самосинхронная схема разряда регистра сдвига [10] (рис. 4).

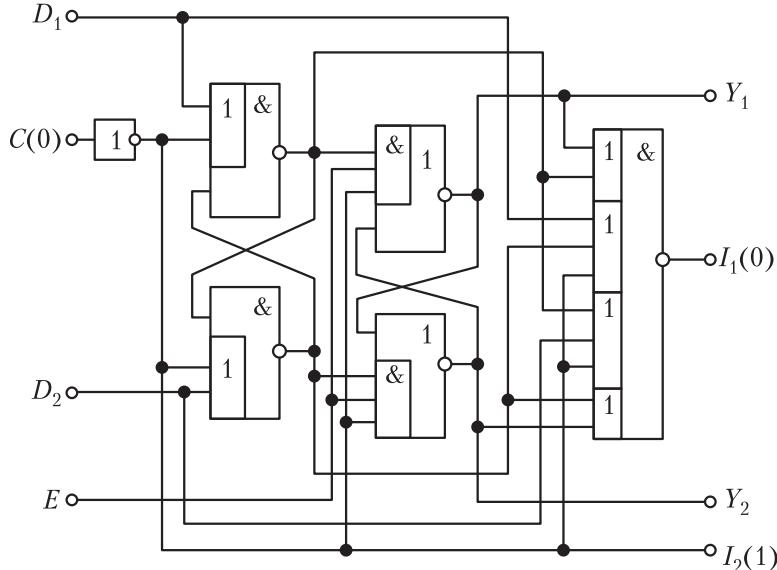


Рис. 4 Разряд регистра сдвига: D_1 и D_2 — входные БИН-сигналы; Y_1 и Y_2 — выходные БИН-сигналы; C и E — У-сигналы; I_1 и I_2 — И-сигналы; в скобках — значения спейсеров

На верхний уровень будет передаваться следующая информация.

В спайсере на выходе I_1 индицируются сигналы C , Y_1 , Y_2 и I_2 , на выходе I_2 — сигнал C . В рабочей фазе на выходе I_1 индицируются сигналы C и I_2 , на выходе I_2 — сигнал C .

Для БИН-входов D_1 и D_2 блок-функция $F = \hat{I}_2$, внешний блокирующий сигнал — C с задержкой, а также выходные сигналы I_2 и I_1 .

Для БИН-выходов Y_1 и Y_2 блокирующая функция $F = EI_2$, а внешние блокирующие сигналы — C и E .

Пример 4. Самосинхронный триггер с усиленными выходами [11] (рис. 5).

На верхний уровень будет передаваться следующая информация.

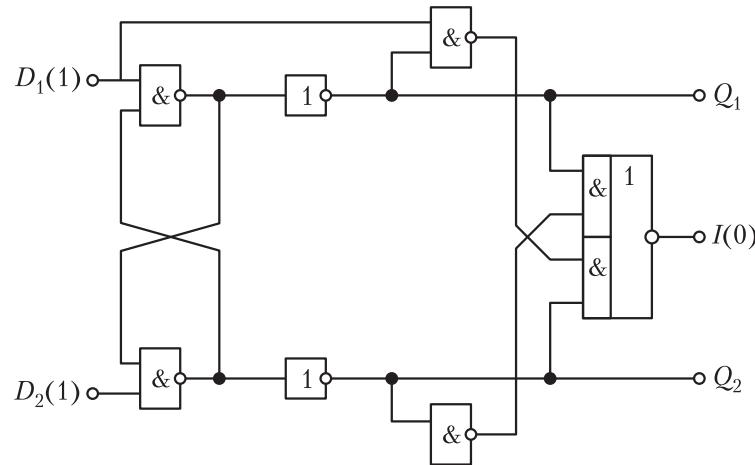


Рис. 5 Одноступенчатый триггер с усиленными выходами: D_1 и D_2 — ПФС-сигналы; Q_1 и Q_2 — БИН-сигналы; I — И-сигнал; в скобках — значения спейсеров

В спейсере на выходе I индицируются сигналы D_1 и D_2 , в рабочей фазе на нем индицируются сигналы D_1 , D_2 , Q_1 и Q_2 .

Для выходных БИН-сигналов Q_1 и Q_2 внешними блок-сигналами будут D_1 и D_2 .

5 Заключение

Одной из главных трудностей проектирования СС-схем практических размеров является анализ самосинхронности. Во всех случаях требуется вычислить и проверить все рабочие состояния схемы и переходы между ними.

Существующие событийные методы оперируют уравнениями всей схемы целиком. При увеличении размера схемы вычислительные затраты на анализ этими методами становятся неприемлемыми, что не позволяет анализировать схемы все увеличивающихся размеров.

В предложенном ранее функциональном иерархическом методе по уравнениям элементов анализируются только фрагменты нижнего уровня. На следующих уровнях используются взаимосвязи фрагментов и информация, полученная на нижележащем уровне.

На нижнем уровне сложность вычислений полиномиальна порядка 2 от числа элементов анализируемой схемы и зависит от числа ее информационных входов M и числа триггеров P как $2^{2(M+P)}$.

Такая зависимость показывает путь рационального разбиения большой схемы на СС-фрагменты. Во фрагментах целесообразно иметь небольшое число ин-

формационных входов и внутренних триггеров, а других элементов может быть практически любое количество.

Литература

1. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под ред. В. И. Варшавского. — М.: Наука, 1986. 400 с.
2. Плеханов Л. П., Степченков Ю. А. Экспериментальная проверка некоторых свойств строго самосинхронных схем // Системы и средства информатики, 2006. Вып. 16. С. 476–485.
3. Соколов И. А., Степченков Ю. А., Бобков С. Г. и др. Базис реализации супер-ЭВМ эксафлопного класса // Информатика и её применения, 2014. Т. 8. Вып. 1. С. 2–11.
4. Muller D. E., Bartky W. C. A theory of asynchronous circuits // Symposium (International) on the Theory of Switching Proceedings. — Harvard University Press, 1959. Part 1. P. 204–243.
5. Варшавский В. И., Кишиневский М. А., Кондратьев А. Ю., Розенблюм Л. Я., Таубин А. Р. Модели для спецификации и анализа процессов в асинхронных схемах // Техническая кибернетика, 1988. № 2. С. 171–190.
6. Плеханов Л. П. Полнота анализа электронных схем на самосинхронность // Системы и средства информатики, 2010. Вып. 20. № 1. С. 48–58.
7. Плеханов Л. П. Проектирование самосинхронных схем: функциональный подход // Проблемы разработки перспективных микро- и наноэлектронных систем: Сб. науч. тр. IV Всеросс. науч.-технич. конф. (МЭС-2010). — М.: ИППМ РАН, 2010. Ч. 1. С. 424–429.
8. Плеханов Л. П. Основы самосинхронных электронных схем. — М.: Бином. Лаборатория знаний, 2013. 208 с.
9. ГОСТ Р 53480-2009. Надежность в технике. Термины и определения. — М.: Стандартинформ, 2010. 11 с.
10. Степченков Ю. А., Дьяченко Ю. Г., Рождественский Ю. В., Рождественскне А. В. Разряд самосинхронного регистра сдвига. Патент на изобретение 2319232 (РФ). Приоритет от 10.03.2008.
11. Sokolov I. A., Stepchenkov Y. A., Dyachenko Y. G. Self-timed RS-trigger with the enhanced noise immunity. International Application Number PCT/RU/2010/000279. International publication date 28.05.2010.

Поступила в редакцию 16.03.16

SELF-TIMING ANALYSIS OF ELECTRONIC CIRCUITS ON THE LOWER LEVEL OF HIERARCHY

L. P. Plekhanov

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: Self-timed circuits (independent on element's delay) have the unique properties of a lack of competitions and safe on Out-Stack-At-Fault (OSAF). They require analysis on self-timing. In the traditional approach — analyzing of elements switching, computational complexity is so great that it does not allow analyzing the most practical circuits. The functional hierarchical method, previously proposed by the author, analyzes logic equations only at the lower level, and at the upper levels, it examines only the relationships between blocks. The suggested method makes it possible to analyze circuits of any size effectively. This article describes in detail this method at the lower level of the hierarchy.

Keywords: self-timed circuits; asynchronous circuits; self-timing analysis; hierarchical analysis

DOI: 10.14357/08696527160202

Acknowledgments

The research was performed under partial financial support of the Program of Fundamental Research 2016 of the Presidium of RAS (project 0063-5015-0015 RAS 1.33P) and subprogram No. 4 of the RAS Department for Nanotechnologies and Information Technologies (ONIT) for 2016 (project 0063-2015-0016 III.3).

References

1. Varshavsky, V. I., ed. 1986. *Avtomatnoe upravlenie asinkhronnymi protsessami v EVM i diskretnykh sistemakh* [Automata control of asynchronous processes in computers and discrete systems]. Moscow: Nauka. 400 p.
2. Plekhanov, L. P., and Yu. A. Stepchenkov. 2006. Eksperimental'naya proverka nekotorykh svoystv strogo samosinkronnykh skhem [Experimental verification of some properties of strictly self-timed circuits]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 16:476–485.
3. Sokolov, I. A., Yu. A. Stepchenkov, S. G. Bobkov, et al. 2014. Bazis realizatsii super-EVM eksaflopnogo klassa [The implementation basis of supercomputer of exaflop class]. *Informatika i ee Primeneniya — Inform. Appl.* 8(1):2–11.
4. Muller, D. E., and W. C. Bartky. 1959. A theory of asynchronous circuits. *Symposium (International) on the Theory of Switching Proceedings*. Harvard University Press. 1:204–243.

5. Varshavskiy, V. I., M. A. Kishinevskiy, A. Yu. Kondrat'ev, L. Ya. Rozenblyum, and A. R. Taubin. 1988. Modeli dlya spetsifikatsii i analiza protsessov v asinkhronnykh skhemakh [Models for specification and analysis of processes in asynchronous circuits]. *Tekhnicheskaya Kibernetika* [Technilal Cybernetics] 2:171–190.
6. Plekhanov, L. P. 2010. Polnota analiza elektronnykh skhem na samosinkhronnost' [The completeness of analysis of electronic circuits for self-timing]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 20(1):48–58.
7. Plekhanov, L. P. 2010. Proektirovanie samosinkhronnykh skhem: funktsional'nyy podkhod [Designing of self-timed circuits: A functional approach]. *Sb. nauch. tr. 4th Vseross. nauch.-tekhnich. konf. "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES-2010)"* [4th All-Russian Scientific-Technical Conference "Problems of Development of Perspective Micro and Nanoelectronic Systems (MES-2010)" Proceedings]. Moscow. 1:424–429.
8. Plekhanov, L. P. 2013. *Osnovy samosinkhronnykh elektronnykh skhem* [The basis of self-timed electronic circuits]. Moscow: Binom. Laboratoriya znaniy. 208 p.
9. GOST R 53480-2009. 2010. Nadezhnost' v tekhnike. Terminy i opredeleniya [Reliability in technique. Terms and definitions]. Moscow: Standartinform Publs. 11 p.
10. Stepchenkov, Yu. A., Yu. G. D'yachenko, Yu. V. Rozhdestvenskiy, and A. V. Rozhdestvenskene. 2008. Razryad samosinkhronnogo registra sdvigа [Unit of self-timed shift register]. Patent RF No. 2319232.
11. Sokolov, I. A., Yu. A. Stepchenkov, and Yu. G. Dyachenko. 28.05.2010. Self-timed RS-trigger with the enhanced noise immunity. International Application No. PCT/RU/2010/000279.

Received March 16, 2016

Contributor

Plekhanov Leonid P. (b. 1943)— Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119133, Russian Federation; lplekhanov@inbox.ru

ЛЕКСИЧЕСКИЙ АНАЛИЗ ДИНАМИЧЕСКИ ФОРМИРУЕМЫХ СТРОКОВЫХ ВЫРАЖЕНИЙ*

М. И. Полубелова¹, С. В. Григорьев²

Аннотация: Строковые выражения могут использоваться для формирования и последующего исполнения кода во время выполнения основной программы. Такой подход обладает высокой выразительностью, однако затрудняет разработку, отладку и сопровождение, а также является источником таких уязвимостей, как внедрение SQL (Structural Query Language) и межсайтовый скрипting. Статический анализ строковых выражений предназначен для борьбы с недостатками подхода посредством проверки того, что все формируемые выражения удовлетворяют некоторым свойствам, без запуска программы. Лексический анализ, или токенизация формируемого кода, является важным шагом такого статического анализа. В статье описан автоматизированный подход к созданию лексических анализаторов динамически формируемого кода, который позволит упростить создание инструментов, предназначенных для статического анализа такого кода.

Ключевые слова: анализ строковых выражений; генератор лексических анализаторов; лексический анализ; встроенные языки

DOI: 10.14357/08696527160203

1 Введение

Многие языки программирования позволяют работать со строковыми выражениями. Последние могут формироваться динамически с использованием строковых операций и языковых конструкций, например условных операторов и циклов. Такие выражения широко используются в программных интерфейсах ODBC (Open DataBase Connectivity), ADO.NET (ActiveX Data Objects.NET) и JDBC (Java DataBase Connectivity), предназначенных для формирования запросов к базе данных на языках программирования C++, C# и Java соответственно, а также в web-программировании. Некоторые примеры использования динамически формируемых строковых выражений представлены в листинге 1.

Динамически формируемые строковые выражения воспринимаются компилятором как обычные строки, что усложняет разработку и сопровождение системы. Во-первых, о наличии ошибок, таких как лексические или синтаксические,

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ», ТМРА-2015), Санкт-Петербург, 12–14 ноября 2015 г.

¹Санкт-Петербургский государственный университет, polubelovam@gmail.com

²Санкт-Петербургский государственный университет, Semen.Grigorev@jetbrains.com

```
<?php
//Embedded SQL
$query = 'SELECT * FROM '. $my_table;
$res = mysql_query($query);
//HTML markup generation
echo "<table>\n";
while($line=mysql_fetch_array($res, MYSQL_ASSOC))
{
    echo "\t<tr>\n";
    foreach ($line as $col_value)
        {echo "\t\t<td>$col_value</td>\n";}
    echo "\t</tr>\n";
}
echo "</table>\n";
?>
```

Листинг 1 Использование нескольких встроенных в PHP языков

в сформированном выражении становится известно только в момент выполнения программы, когда оно начинает выполняться в своем программном окружении. Во-вторых, при ненадлежащей обработке пользовательского ввода система становится уязвимой, например для SQL-инъекций или межсайтового скрипtingа. Указанные проблемы можно решить, включив обработку строковых выражений в статический анализ программы.

Классический подход к статическому анализу заключается в проведении лексического анализа и синтаксического разбора исходного кода. Синтаксический анализ строит структурное представление, которое используется в дальнейшем, например, для семантического анализа или трансформаций, проводимых в контексте реинжиниринга. В рамках данного подхода широко распространен автоматизированный способ создания лексических и синтаксических анализаторов такими генераторами, как Lex, Yacc и их потомками. Использование подобных инструментов сокращает затраты на создание программных продуктов, требующих построения структурного представления кода. Однако существующие генераторы лексических и синтаксических анализаторов не применимы для создания инструментов обработки динамически формируемого кода из-за того, что такой код, как правило, не представим в виде линейного потока, принимаемого на вход классическими анализаторами.

Таким образом, есть необходимость в создании генератора лексических анализаторов для динамически формируемого кода, который предоставляет функциональность, аналогичную классическому. Это позволит упростить создание инструментов, которые предназначены для решения задач, возникающих при обработке динамически формируемого кода. Примерами таких задач являются трансформация запросов с одного языка на другой, возникающая в контексте реинжиниринга информационных систем, подсветка синтаксиса и ошибок в интегрированных средах разработки, а также подсчет различных метрик.

В данной статье описан автоматизированный подход к созданию лексического анализатора для динамически формируемого кода. В рамках работы был разработан алгоритм лексического анализа и генератор лексических анализаторов, за основу которого была взята библиотека FsLex¹. Указанные компоненты были реализованы как часть проекта YaccConstructor², который служит модульным инструментом для проведения лексического анализа и синтаксического разбора, а также платформой для поддержки встроенных языков.

2 Обзор

В данном разделе рассматриваются существующие инструменты, предназначенные для работы с динамически формируемыми выражениями, генератор лексических анализаторов FsLex и проект YaccConstructor, в котором ведется разработка автоматизированного подхода к созданию лексических анализаторов для динамически формируемого кода.

2.1 Обзор существующих инструментов

Для работы с динамически формируемыми строковыми выражениями существует ряд инструментов. Почти все они предназначены для решения какой-то одной конкретной задачи: либо проверки выражения на соответствие описанию некоторой эталонной грамматики, либо статического анализа программы на уязвимость. Так как генерация всех значений динамически формируемого выражения значительно снижает скорость проведения анализа и возможны ситуации, когда число принимаемых выражением значений может быть бесконечным, то целесообразно иметь конечное представление множества значений данного выражения и уже над ним проводить анализ.

Конечное представление множества значений строкового выражения впервые было использовано в инструменте Java String Analyzer [1]. Этот инструмент предназначен для анализа строк и строковых операций в Java-программе. Результатом этого приближения стал конечный автомат, который используется для проверки включения языков: проверяется включение языка, порождаемого программой, в язык, описанный пользователем. Затем инструмент PHP String Analyzer [2], используя идею предыдущего инструмента, уточнил проводимую аппроксимацию, результатом которой стала контекстно-свободная грамматика. Этот инструмент применяется для статической валидации HTML-страниц, генерируемых в PHP-программе.

В инструменте Pixy [3], предназначенном для поиска SQL-инъекций и межсайтового скрипtingа в PHP-программах, применяется техника path pruning, позволяющая проводить анализ только тех значений строкового выражения,

¹<http://fsprojects.github.io/FsLexYacc/>.

²<https://github.com/YaccConstructor/YaccConstructor>.

которые оно может принять в момент выполнения программы. Инструмент Stranger [4] расширяет указанный подход, используя в качестве структурного представления динамически формируемого кода конечный автомат над алфавитом символов обрабатываемого языка. При этом рассматривается общий случай, когда аргументами строковых операций являются конечные автоматы. В инструменте Stranger разработан алгоритм, который вычисляет результат этих операций и возвращает его в виде конечного автомата, что позволяет достичь более высокой точности анализа по сравнению с аналогами.

Разработчики следующего инструмента расширили круг решаемых задач, сформулировав вопросы безопасности, корректности и производительности сформированных запросов с использованием таких программных интерфейсов, как ADO.NET и JDBC. Описание этого инструмента дается в статье [5], в которой также была указана разработанная функциональность: поиск SQL-инъекций, извлечение множества всех значений для строкового выражения, удаление неиспользуемых переменных в формируемом запросе, а также проверка на соответствие типов возвращаемого запросом значения ожидаемым в программе.

Инструмент SAFELI [6], также предназначенный для поиска уязвимостей в веб-приложениях, отличается от рассмотренных тем, что структурным представлением динамически формируемого кода является синтаксическое дерево разбора. Результат получается посредством сопоставления полученного дерева с синтаксическим деревом шаблона уязвимости, параметризованного реальными данными. Однако SAFELI не поддерживает обработку строковых выражений, которые могут быть получены при участии строковых операций.

Для проведения лексического анализа и синтаксического разбора множества значений строкового выражения был разработан инструмент Alvor [7], который является расширением для среды разработки Eclipse, предназначенным для статической валидации SQL-выражений, встроенных в программы на Java. Одна из возможностей инструмента — поиск лексических и синтаксических ошибок, однако поддержка нового языка генерируемого кода является нетривиальной задачей из-за отсутствия генераторов лексических и синтаксических анализаторов. Кроме того, Alvor не поддерживает обработку выражений, полученных с помощью строковых операций (кроме конкатенации) и циклов.

2.2 Инструмент YaccConstructor

YaccConstructor [8] является модульным инструментом с открытым исходным кодом, предназначенным для исследований в области лексического анализа и синтаксического разбора, а также платформой для поддержки встроенных языков [9]. Данный инструмент реализован на платформе Microsoft .NET, основной язык разработки — F#.

Разработанный механизм анализа встроенных языков ранее имел ограничения на структуру динамически формируемого выражения: лексический и синтаксический анализаторы могли обрабатывать только аппроксимацию, представленную

в виде ориентированного ациклического графа. Это не позволяло корректно обрабатывать выражения, полученные с помощью циклов.

В данной работе такое ограничение снимается: лексический анализатор работает с произвольным конечным автоматом над алфавитом символов обрабатываемого языка. Разработанный модуль для лексического анализа, который состоит из генератора лексических анализаторов и интерпретатора, соответствующего предложенному алгоритму лексического анализа, внедрен в инструмент YaccConstructor.

2.3 Генератор лексических анализаторов FsLex

При проведении лексического анализа часто используются генераторы лексических анализаторов, которые по спецификации обрабатываемого языка строят описание конечного преобразователя, на основе которого входной поток символов преобразуется в поток токенов. В качестве инструмента для проведения лексического анализа динамически формируемого кода был выбран генератор лексических анализаторов FsLex. Этот выбор обусловлен тем, что реализованный механизм является компонентом инструмента YaccConstructor, основным языком разработки которого служит язык программирования F#.

Генератор лексических анализаторов на вход принимает файл с расширением .fsl, в котором описана лексическая спецификация языка, формат определения которой представлен в листинге 2.

Результатом работы генератора является файл с расширением .fs с кодом F# для лексического анализатора. Этот файл содержит код, указанный в заголовке спецификации, конечный преобразователь и функции, которые были созданы на каждую точку вхождения, а также вызов функции интерпретатора

```
{  
module Lexer  
// header: any valid F# code can appear here  
open Parser //specifies type of tokens  
}  
// regex macros  
let ident = regexp ...  
// rules  
rule entrypoint = parse  
| regexp { action }  
| ...  
and entrypoint = parse  
...
```

Листинг 2 Формат определения спецификации для языка

построенного конечного преобразователя. Чтобы использовать такой лексический анализатор, полученный файл вместе с описанием типов токенов, которые автоматически строятся по грамматике эталонного языка, необходимо добавить в модуль, предназначенный для лексического анализа.

3 Лексический анализ

Основная задача лексического анализа — преобразование входного потока символов в поток токенов, соответствующих спецификации обрабатываемого языка, и сохранение привязки лексических единиц к исходному коду. В классическом случае входной поток является линейным. Для проведения лексического анализа динамически формируемого выражения необходима структура, которая выступает конечным представлением множества значений этого выражения. Для построения аппроксимации используется алгоритм, предложенный в статье [10], поэтому такой структурой служит конечный автомат над алфавитом символов обрабатываемого языка.

Результатом работы лексического анализа динамически формируемого строкового выражения является конечный автомат над алфавитом токенов эталонной грамматики языка. В классическом лексическом анализе токен можно представить в виде структуры, содержащей идентификатор токена и последовательность символов, выделенных из входного потока. В контексте данной статьи токен представляет собой структуру, содержащую идентификатор токена и *конечный автомат*, описывающий все возможные последовательности символов для данного токена в данной позиции. При этом для каждого символа хранится информация: из какой строки получен этот символ и координаты его позиций внутри этой строки. Это необходимо для того, чтобы сохранить информацию о происхождении токена, так как он мог быть сформирован из различных строковых переменных в исходном коде.

Таким образом, **основная задача лексического анализа динамически формируемого строкового выражения** заключается в переводе конечного автомата над алфавитом символов обрабатываемого языка в конечный автомат над алфавитом токенов эталонной грамматики языка с сохранением привязки лексических единиц к исходному коду.

В данной статье для конечных автоматов и конечных преобразователей используются определения, представленные ниже.

Конечным автоматом называется кортеж $A = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где Q — конечное множество состояний; Σ — входной алфавит; Δ задает на множестве Q структуру ориентированного графа, дуги которого помечены символами (x) , где $x \in \Sigma \cup \{\varepsilon\}$; q_0 — начальное состояние; $F \subseteq Q$ — множество конечных состояний.

Конечным преобразователем называется кортеж $M = \langle Q, \Sigma, \Sigma', \Delta, q_0, F \rangle$, где Q — конечное множество состояний; Σ и Σ' — входной и выходной алфавиты соответственно; Δ задает на множестве Q структуру ориентированного графа,

дуги которого помечены парами $(x : y)$, где $x \in \Sigma$ и $y \in \Sigma' \cup \{\varepsilon\}$; q_0 — начальное состояние; $F \subseteq Q$ — множество конечных состояний.

В алгоритме лексического анализа используется операция **композиции** над двумя конечными преобразователями. Композиция конечных преобразователей [11] — это два последовательно взаимодействующих конечных преобразователя: выход первого конечного преобразователя служит входом для второго конечного преобразователя.

3.1 Генератор лексических анализаторов

Для проведения лексического анализа динамически формируемого выражения к двум конечным преобразователям применяется операция композиции, которая использует явное представление этих преобразователей, что порождает ограничения на формат определения лексической спецификации для языка.

Генератор лексических анализаторов FsLex строит конечный преобразователь, в котором входным алфавитом являются символы, имеющие кодировку ASCII или Unicode, выходным алфавитом — функции, тип возвращаемых значений которых соответствует типу `Token`. Эти функции соответствуют действиям (`action`), которые определены в спецификации для языка. Однако бывают ситуации, когда нужно исключить некоторые выражения, например пробелы и комментарии, из результата. Обычно это происходит на этапе проведения лексического анализа: в соответствующем действии не происходит возвращения токена, однако возвращаемое значение должно иметь тип `Token`. В классическом случае происходит вызов функции, соответствующей точке вхождения, от измененного состояния буфера лексического анализатора. Такой способ не применим для лексического анализа динамически формируемого кода, когда используется операция композиции, из-за ограничений на выходной алфавит. В данной работе предлагается использование следующих типов `Option`: `Some` обозначает возвращение токена; `None` — его отсутствие.

Использование нескольких точек вхождения в определении спецификации означает рекурсивное определение функций, которые создаются на каждую точку вхождения: в соответствующем действии происходит вызов одной из этих функций. Обычно такой подход используют для обработки вложенных конструкций, например комментариев. Данный случай не учитывается в реализованном инструменте, что создает ограничение на число используемых точек вхождения: можно использовать только одну точку вхождения.

Чтобы показать введенные ограничения, в листингах 3 и 4 указана спецификация для языка арифметических выражений, которую принимают на вход генератор FsLex и разработанный инструмент соответственно.

В результате своей работы генератор создает файл с расширением `.fs` с кодом на языке F# для лексического анализатора. В этом файле содержится код, указанный в заголовке спецификации, конечный преобразователь, массив действий и функция `tokenize`. Функция `tokenize` осуществляет лексический разбор.

```
rule token = parse
| whitespace { token lexbuf }
| ['-']? digit+ ('.' digit+)??
| ['e' 'E'] digit+?
| { NUMBER(lexbuf) }
| '-'
| { MINUS(lexbuf) }
| '/'
| { DIV(lexbuf) }
| '+'
| { PLUS(lexbuf) }
| "**"
| { POW(lexbuf) }
| '*'
| { MULT(lexbuf) }
```

Листинг 3 FsLex

```
rule token = parse
| whitespace { None }
| ['-']? digit+ ('.' digit+)??
| ['e' 'E'] digit+?
| { Some(NUMBER(gr)) }
| '-'
| { Some(MINUS(gr)) }
| '/'
| { Some(DIV(gr)) }
| '+'
| { Some(PLUS(gr)) }
| "**"
| { Some(POW(gr)) }
| '*'
```

Листинг 4 YaccConstructor

```
[| (fun (gr:FSA<_>) -> None );
  (fun (gr:FSA<_>) -> Some(NUMBER(gr)));
  (fun (gr:FSA<_>) -> Some(MINUS(gr)));
  (fun (gr:FSA<_>) -> Some(DIV(gr)));
  (fun (gr:FSA<_>) -> Some(PLUS(gr)));
  (fun (gr:FSA<_>) -> Some(POW(gr)));
  (fun (gr:FSA<_>) -> Some(MULT(gr))); |]
```

Листинг 5 Массив действий для спецификации языка, указанного в листинге 4

Массив действий состоит из функций, задающих отображение из некоторого конечного автомата в тип `Option<'token>`. Для спецификации языка из листинга 4 массив действий представлен в листинге 5.

3.2 Алгоритм лексического анализа

На этапе построения аппроксимации множества значений динамически формируемого строкового выражения происходит сохранение привязки: с каждым символом сохраняются координаты позиций этого символа в исходной строке, а также привязка этой строки к исходному коду. Результатом этого этапа является конечный автомат A , в котором входной алфавит состоит из элементов вида `('символ', привязка)`.

Над конечным автоматом A запускается процедура построения детерминированного конечного автомата. При этом конечный автомат детерминирован так, чтобы на дугах из одной вершины не было двух одинаковых символов с одинаковой привязкой. Из полученного конечного автомата $A' = \langle Q, \Sigma, \Delta', q_0, F \rangle$ строится конечный преобразователь $M = \langle Q, \Sigma, \Sigma', \Delta, q_0, F \rangle$ для лексического анализатора, в котором выходной алфавит $\Sigma' = \{s | (s, _) \in \Sigma\}$, т. е. дуги графа, являющиеся представлением конечного преобразователя M , помечены парами `(('символ', привязка) : 'символ')`. Отображение Δ отличается от отобра-

жения Δ' только тем, что появился выходной алфавит. Такое преобразование необходимо для выполнения операции композиции.

В конечный преобразователь M необходимо добавить переход по символу '`eof`' из всех конечных состояний в новое состояние, которое теперь станет конечным. Этот символ означает окончание строки и необходим для корректной работы лексического анализатора, так как вычисление действия (*action*) к накопленной строке происходит при чтении следующего символа.

На вход лексический анализатор принимает два конечных преобразователя, один из которых получен в результате построения аппроксимации, а второй — из описания, построенного генератором лексических анализаторов. Предлагаемый алгоритм для проведения лексического анализа динамически формируемого выражения состоит из двух этапов.

Этап 1. Выполнение операции композиции над двумя входными конечными преобразователями. Результатом этой операции является либо набор лексических ошибок, либо конечный преобразователь $M_1 = \langle Q_1, \Sigma_1, \Sigma'_1, \Delta_1, q_01, F_1 \rangle$. Наличие лексических ошибок возможно в двух случаях: либо конечный преобразователь M содержит символы, которых нет в лексической спецификации, либо конечный преобразователь M порождает такой язык, который не принимает на вход лексический анализатор. Возможна также ситуация, когда конечный преобразователь M порождает язык, в котором есть слова, не принимаемые на вход лексическим анализатором.

Этап 2. Если конечный преобразователь M_1 получен, то происходит этап его интерпретации, результатом которой станет конечный автомат A_{token} над алфавитом токенов. В конечном преобразователе M_1 выделяются *action-вершины* — это вершины, из которых выходит хотя бы одна дуга, содержащая функцию, возвращающую `Some(token)`. После *action-вершины* всегда возвращается один тип токена, но может выходить дуга, помеченная символом $(-, \varepsilon)$ и означающая, что токен продолжает накапливаться. Пример такого конечного преобразователя представлен на рис. 1. Из одной *action-вершины* могут быть достижимы несколько других *action-вершин*. Число достижимых вершин соответствует числу токенов, которые нужно выделить в конечном преобразователе M_1 для данной *action-вершины*.

Этот этап состоит из шагов, представленных ниже. На них используется структура `GraphAction`, которая предназначена для выделения токенов в конечном преобразователе M_1 и содержит поля для стартовой вершины, набора конечных вершин и конечного автомата. Стартовой вершиной может быть только *action-вершина* или начальное состояние, конечной вершиной — *action-вершина* или конечное состояние.

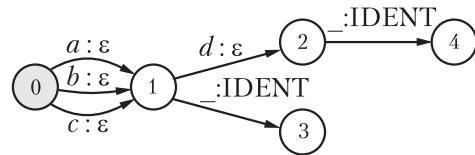


Рис. 1 Вершины 1 и 2 являются *action-вершинами*, но из вершины 1 токен продолжает накапливаться

В описании алгоритма для дуги конечного преобразователя M_1 используется нотация $e = (u, (a, b), v)$, где u — начальная вершина дуги; v — конечная вершина; (a, b) — метка дуги; $a \in \Sigma_1$, $y \in \Sigma'_1 \cup \{\varepsilon\}$. Добавление дуги в конечный автомат, который содержится в структуре `GraphAction`, осуществляется с помощью функции `graphAction.fsa.AddEdge(e)`, которая добавляет дугу из вершины u в вершину v , помеченную символом a , в соответствующий конечный автомат. В алгоритме используются очередь Q , для того чтобы контролировать порядок обхода конечного преобразователя, и структура `GraphAction grAct` для выделения конечного автомата, который содержит токен.

Шаг 1. Для конечного преобразователя M_1 строится набор вершин `actV`, которые являются action-вершинами.

Шаг 2. Для начального состояния конечного преобразователя M_1 и вершин из набора `actV` запускается обход, который накапливает для токена конечный автомат. Порядок обхода представлен в алгоритме 1. Результатом обхода

Алгоритм 1 Порядок обхода конечного преобразователя для сохранения привязки лексических единиц к исходному коду

```
function BFS(vStart, M1)
    grAct.startV ← vStart
    Q.Enqueue(vStart)
    while Q is not empty do
        v ← Q.Dequeue()
        if v is not visited then
            VISIT(v)
            for all e = (v, _, u) in M1 do
                if e = (v, (_, ε), u) then
                    c ← v is init state of M1
                    if v ≠ vStart or c then
                        grAct.fsa.AddEdge(e)
                        if u = vStart then
                            grAct.endV.Add(u)
                            Q.Enqueue(u)
                        else
                            if v = vStart then
                                grAct.fsa.AddEdge(e)
                                if u is final state in M1 then
                                    grAct.endV.Add(u)
                                else
                                    Q.Enqueue(u)
                            else
                                grAct.endV.Add(v)
```

Алгоритм 2 Порядок обхода инвертированного конечного преобразователя для сохранения привязки лексических единиц к исходному коду

```
function BFSINV (vStart,  $M_2$ ,  $M_1$ )
    grAct.startV  $\leftarrow$  vStart
    Q.Enqueue(vStart)
    while  $Q$  is not empty do
         $v \leftarrow Q$ .Dequeue()
        if  $v$  is not visited then
            VISIT( $v$ )
            for all  $e = (v, \_, u)$  in  $M_2$  do
                grAct.fsa.AddEdge( $e$ )
                 $c \leftarrow u$  is action-vertex or init state of  $M_2$ 
                if  $c$  then
                    if  $\exists e = (u, (\_, \varepsilon), \_)$  in  $M_1$  then
                        Q.Enqueue( $u$ )
                    else
                        Q.Enqueue( $u$ )
```

для всех вершин является коллекция `tokenAct`, состоящая из элементов типа `GraphAction`.

Шаг 3. Для точного определения конечного автомата, сохраняющего связь между токеном и исходным кодом, необходимо пройти конечный преобразователь M_1 в обратном направлении. Такая необходимость возникает в случае, если конечный преобразователь M_1 содержит циклы. На этом шаге также запускается обход для вершин из набора `actV` и начального состояния конечного преобразователя M_2 , который является инвертированным конечным преобразователем M_1 . Порядок обхода представлен в алгоритме 2. Результатом обхода для всех вершин является коллекция `tokenActInv`, состоящая из элементов типа `GraphAction`.

Шаг 4. Ищутся пересечения конечных автоматов, полученных на двух предыдущих шагах, при условии, что `tokenAct.endV.current = tokenActInv.startV`. Токен определяется функцией, лежащей на дуге, исходящей из вершины `tokenAct.endV.current`, в него записывается результат пересечения. При этом в конечный автомат A_{token} добавляется переход по этому токену из вершины `tokenAct.startV` в вершину `tokenAct.endV.current`.

Так как символ '`'eof'`' является вспомогательным символом для лексического анализа, то для всех вершин, из которых выходит дуга, помеченная '`'eof'`', добавляется переход в новое состояние по токену `EOF` в конечный автомат A_{token} , которое теперь становится конечным состоянием A_{token} . Над полученным конечным автоматом запускается процедура построения детерминированного конечного автомата. При этом автомат детерминирован так, чтобы на дугах из одной вершины не было двух токенов с одинаковым идентификатором и конечным автоматом.

3.3 Пример

Рассмотрим работу алгоритма лексического анализа на примере. Пусть результатом аппроксимации является конечный автомат A , представленный на рис. 2, *a*. Преобразуем этот конечный автомат во входную структуру для алгоритма. Для этого сперва строится детерминированный конечный автомат A' (рис. 2, *б*), затем конечный преобразователь M (рис. 2, *в*).

Для обработки конечного преобразователя M используется спецификация для языка арифметических выражений (см. листинг 4). Результат композиции конечного преобразователя M с конечным преобразователем, построенным генератором лексических анализаторов, показан на рис. 3.

На рис. 3 цифры соответствуют индексам в массиве действий, которые представлены в листинге 5 (нумерация элементов в массиве начинается с 0). Индекс 4 соответствует функции, которая возвращает токен `Some(PLUS)`, 5 — токен `Some(POW)`, 6 — токен `Some(MULT)`. Так как лексических ошибок получено не было, то происходит этап интерпретации полученного конечного преобразователя M_1 , который состоит из 4 шагов. Ниже представлены результаты выполнения каждого шага.

Шаг 1. В набор `actV` добавляются action-вершины 1, 2 и 3.

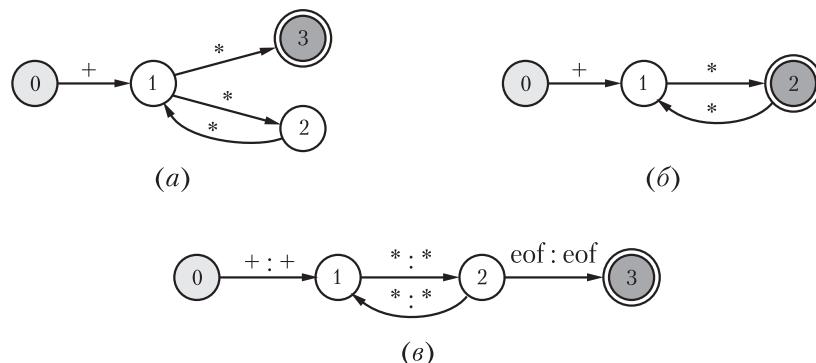


Рис. 2 Получение конечного преобразователя для лексического анализа: (*а*) конечный автомат A ; (*б*) конечный автомат A' ; (*в*) конечный преобразователь M

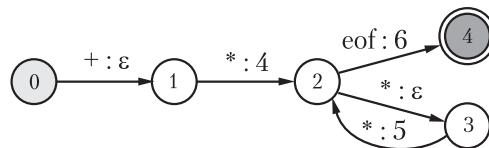


Рис. 3 Результат композиции

Таблица 1 Коллекция `tokenAct` для шага 2

startV	FSA	endV
0		1
1		2,3
2		2,3
3		4

Шаг 2. Запускается обход, представленный в алгоритме 1, для вершин из набора `actV` и начального состояния конечного преобразователя M_1 (вершина 0). Результатом обхода для всех вершин является коллекция `tokenAct`, которая представлена в виде табл. 1.

Шаг 3. Запускается обход, представленный в алгоритме 2, для вершин из набора `actV` и начального состояния конечного преобразователя M_2 (вершина 49). Результатом обхода для всех вершин является коллекция `tokenActInv`, которая представлена в виде табл. 2.

Шаг 4. Выполняются операции пересечения конечных автоматов, результаты которых представлены в табл. 3.

Результатом лексического анализа выступает конечный автомат, представленный на рис. 4. Каждая дуга графа содержит токен, который хранит в себе конечный автомат. Например, дуга от вершины 1 к вершине 3 содержит

Таблица 2 Коллекция `tokenActInv` для шага 3

startV	FSA
1	
2	
3	
4	

Таблица 3 Результат пересечения конечных автоматов из табл. 1 и 2

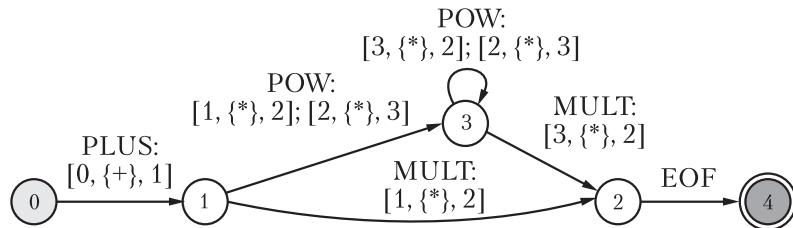


Рис. 4 Результат лексического анализа

токен **POW**, у которого хранится конечный автомат с переходами от состояния 1 к состоянию 2 по символу '*', от 2 к 3 — по символу '*', полученный в результате пересечения конечных автоматов.

4 Архитектура модуля лексического анализа

Архитектура инструмента, реализующего рассмотренный механизм, представлена на рис. 5.

Компонент **Лексический анализатор** состоит из **Генератора лексических анализаторов** и **Интерпретатора**. Генератор лексических анализаторов строит конечный преобразователь, описание которого взято из **Библиотеки для конечных автоматов и конечных преобразователей**, и сохраняет результат в отдельный файл. Такой подход позволяет многократно использовать данный конечный преобразователь для обработки кода, написанного на одном языке. **Интерпретатор** принимает на вход два конечных преобразователя, один из кото-

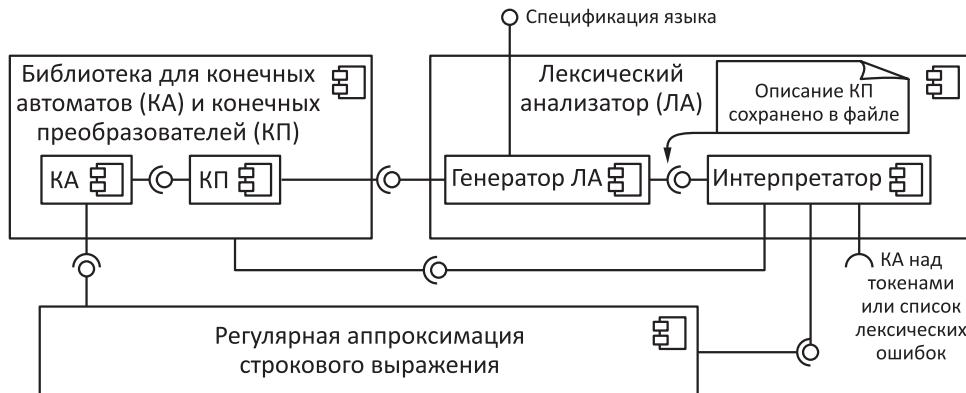


Рис. 5 Архитектура модуля для лексического анализа

рых получен в результате построения аппроксимации (за это отвечает компонент **Регулярная аппроксимация строкового выражения**), а второй построен генератором лексических анализаторов. Результатом работы **Интерпретатора** является либо конечный автомат над алфавитом токенов эталонной грамматики языка, либо список обнаруженных лексических ошибок.

Библиотека для конечных автоматов и конечных преобразователей предоставляет ряд операций, которые необходимы для построения аппроксимации и проведения лексического анализа. Для конечных автоматов используются операции: конкатенация, replace, дополнение, пересечение, а для конечных преобразователей — композиция.

5 Апробация

В данном разделе описан механизм использования реализованного инструмента, а также рассмотрены примеры, демонстрирующие разработанную функциональность.

Для осуществления лексического разбора необходимо выполнить следующие шаги.

Шаг 1. Запустить генератор, указав путь к файлу с расширением .fsl, в котором написана спецификация. В результате создается файл с расширением .fs, содержащий конечный преобразователь и вспомогательные функции.

Шаг 2. Необходимо в отдельном файле с расширением .fs указать описание типов токенов, которые автоматически строятся по грамматике эталонного языка. Полученные файлы подключить к модулю, предназначенному для получения результата лексического разбора.

Шаг 3. Получить конечный автомат, аппроксимирующий множество значений строкового выражения и удовлетворяющий описанию используемой библиотеки для конечных автоматов и конечных преобразователей. Преобразовать этот конечный автомат в конечный преобразователь (см. подразд. 3.2).

Шаг 4. Вызвать функцию `tokenize` из сгенерированного файла. Результатом действия этой функции будет либо конечный автомат над алфавитом токенов эталонной грамматики языка, либо список лексических ошибок.

Рассмотрим примеры, которые показывают преимущества реализованного решения, а именно: возможность сохранения конечного автомата внутри структуры токена и обработки циклов во входном конечном автомате.

Пример 1. Рассмотрим пример кода (листинг 6).

Результатом аппроксимации выражения `query` является конечный автомат, представленный на рис. 6. Результат лексического анализа представлен на рис. 7. В результирующем конечном автомате учитываются две ситуации: цикл не выполняется (путь $0 \rightarrow \dots \rightarrow 7 \rightarrow 9$) и выполняется (путь $0 \rightarrow \dots \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 8 \rightarrow \dots \rightarrow 10 \rightarrow 9$), что позволяет уточнить проводимый анализ в целом.

```
private void Go(int number){
    String query = "SELECT nameX FROM tableY WHERE x < ";
    while(query.Length < number){
        query += "+ 1 ";
    }
    Program.ExecuteImmediate(query);
}
```

Листинг 6 Пример формирования выражения в цикле



Рис. 6 Результат аппроксимации

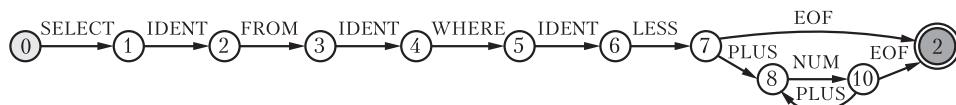


Рис. 7 Результат лексического анализа

Пример 2. Рассмотрим пример кода, в котором конечный автомат токена содержит цикл (листинг 7).

Результатом лексического анализа будет конечный автомат, представленный на рис. 8. Конечный автомат первого токена IDENT, содержащий цикл, представлен на рис. 9. На этом же рисунке показана сохраненная привязка символов к исходному коду. Таким образом, если цикл содержится только в конечном автомате токена, то в результирующем конечном автомате циклы отсутствуют, что позволяет упростить структуру входных данных для дальнейшего анализа.

На практике основным сценарием для динамически формируемого выражения является ситуация, представленная на рис. 10, где строки x_1, x_2, \dots, x_n

```
String query = "SELECT name";
for(int i = 0; i < 10; i++){
    query += "X";
}
query += " FROM tableY";
Program.ExecuteImmediate(query);
```

Листинг 7 Пример кода, в котором конечный автомат токена содержит цикл



Рис. 8 Результат лексического анализа

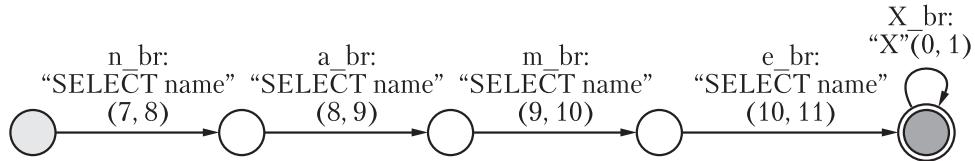


Рис. 9 Конечный автомат первого токена IDENT, содержащий цикл

возвращают одинаковый идентификатор токена. Разработанный механизм лексического анализа вернет один токен при переходе из состояния 1 в состояние 2, что значительно упростит входные данные для синтаксического разбора.

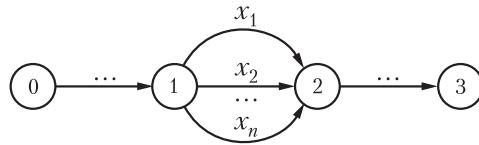


Рис. 10 Конечный автомат

6 Заключение

В данной работе описан алгоритм лексического анализа и основанный на нем генератор лексических анализаторов для динамически формируемого кода. Данный алгоритм был реализован в проекте YaccConstructor на языке программирования F#. Разработанный алгоритм позволяет при проведении лексического анализа для токена сохранять конечный автомат, что значительно упрощает входные данные для синтаксического разбора и привязку лексических единиц к исходному коду, которая необходима для позиционирования места ошибок или навигации по коду. Разработанный генератор лексических анализаторов позволяет получать по спецификации языка соответствующий анализатор, использующий описанный алгоритм. Таким образом, был разработан автоматизированный подход к созданию лексических анализаторов для динамически формируемого кода.

В дальнейшем планируется снять ограничения на формат написания спецификации, по которому генератор лексических анализаторов строит соответствующий анализатор, а также оптимизировать полученный инструмент за счет подбора структур данных и алгоритмов для работы с конечными автоматами [12].

Литература

1. Christensen A. S., Møller A., Schwartzbach M. I. Precise analysis of string expressions // 10th Conference (International) on Static Analysis Proceedings. — Berlin–Heidelberg: Springer-Verlag, 2003. P. 1–18.
2. Minamide Y. Static approximation of dynamically generated web pages // 14th Conference (International) on World Wide Web Proceedings. — New York, NY, USA: ACM, 2005. P. 432–441.
3. Jovanovic N., Kruegel C., Kirda E. Pixy: A static analysis tool for detecting web application vulnerabilities // Symposium on Security and Privacy Proceedings. — Berkeley/Oakland, CA, USA: IEEE, 2006. P. 263–269.
4. Yu F., Alkhalaf M., Bultan T. Stranger: An automata-based string analysis tool for PHP // Tools and algorithms for the construction and analysis of systems / Eds. J. Esparza, R. Majumdar. — Lecture notes in computer science ser. — Berlin–Heidelberg: Springer, 2010. Vol. 6015. P. 154–157.
5. Dasgupta A., Narasayya V., Syamala M. A static analysis framework for database applications // Conference on Computer Software and Applications Proceedings. — IEEE, 2007. P. 87–96.
6. Fu X., Qian K. Safeli: Sql injection scanner using symbolic execution // Workshop on Testing, Analysis, and Verification of Web Services and Applications Proceedings. — New York, NY, USA: ACM, 2008. P. 34–39.
7. Annamaa A., Breslav A., Kabanov J., Vene V. An interactive tool for analyzing embedded SQL queries // Programming languages and systems / Ed. K. Veda. — Lecture notes in computer ser. — Berlin–Heidelberg: Springer, 2010. Vol. 6461. P. 131–138.
8. Кириленко Я. А., Григорьев С. В., Авдюхин Д. А. Разработка синтаксических анализаторов в проектах по автоматизированному реинжинирингу информационных систем // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление, 2013. Вып. 3(174). С. 94–98.
9. Grigorev S., Verbitskaia E., Ivanov A., Polubelova M., Mavchun E. String-embedded language support in integrated development environment // 10th Central and Eastern European Software Engineering Conference in Russia Proceedings. — ACM, 2014. P. 21:1–21:11.
10. Yu F., Alkhalaf M., Bultan T., Ibarra O. H. Automata-based symbolic string analysis for vulnerability detection // Form. Method. Syst. Des., 2014. Vol. 44. No. 1. P. 44–70.
11. Hanneforth T. Finite-state machines: Theory and applications. Unweighted finite-state automata. — Universität Potsdam, 2008. 99 p. http://tagh.de/tom/wp-content/uploads/fsm_unweightedautomata.pdf.
12. Hooimeijer P., Veanes M. An evaluation of automata algorithms for string analysis // 12th Conference (International) on Verification, Model Checking, and Abstract Interpretation Proceedings. — Berlin–Heidelberg: Springer-Verlag, 2011. P. 248–262.

Поступила в редакцию 26.02.16

LEXICAL ANALYSIS OF DYNAMICALLY GENERATED STRING EXPRESSIONS

M. I. Polubelova and S. V. Grigorev

Saint Petersburg State University, 7/9 Universitetskaya Nab., St. Petersburg 199034, Russian Federation

Abstract: There is a class of applications which utilizes the idea of string embedding of one language into another. In this approach, a host program generates string representation of clauses in some external language, which are then passed to a dedicated runtime component for analysis and execution. Despite providing better expressiveness and flexibility, this technique makes the behavior of the system less predictable, complicates maintenance, and is a source of such vulnerabilities as SQL injections and cross-site scripting. Static analysis of strings is intended to minimize the drawbacks of the approach by checking well-formedness of a set of all dynamically-generated clauses at compile-time. Lexical analysis, or tokenization, is an important step of static analysis. The paper presents an automated approach to lexical analyzers construction which simplifies implementation of static analyzers of dynamically generated code.

Keywords: string analysis; lexing; string-embedded language; lexer generator

DOI: 10.14357/08696527160203

References

1. Christensen, A. S., A. Møller, and M. I. Schwartzbach. 2003. Precise analysis of string expressions. *10th Conference (International) on Static Analysis Proceedings*. Berlin-Heidelberg: Springer-Verlag. 1–18.
2. Minamide, Y. 2005. Static approximation of dynamically generated web pages. *14th Conference (International) on World Wide Web Proceedings*. New York, NY: ACM. 432–441.
3. Jovanovic, N., C. Kruegel, and E. Kirda. 2006. Pixy: A static analysis tool for detecting web application vulnerabilities. *Symposium on Security and Privacy Proceedings*. Berkeley/Oakland, CA: IEEE. 263–269.
4. Yu, F., M. Alkhafaf, and T. Bultan. 2010. Stranger: An automata-based string analysis tool for PHP. *Tools and algorithms for the construction and analysis of systems*. Eds. J. Esparza, and R. Mayumdar. Lecture notes in computer science ser. Berlin-Heidelberg: Springer. 6015:154–157.
5. Dasgupta, A., V. Narasayya, and M. Syamala. 2007. A static analysis framework for database applications. *Conference on Computer Software and Applications Proceedings*. IEEE. 87–96.
6. Fu, X., and K. Qian. 2008. Safeli: Sql injection scanner using symbolic execution. *Workshop on Testing, Analysis, and Verification of Web Services and Applications Proceedings*. New York, NY: ACM. 34–39.

7. Annamaa, A., A. Breslav, J. Kabanov, and V. Vene. 2010. An interactive tool for analyzing embedded SQL queries. *Programming languages and systems*. Ed. K. Veda. Lecture notes in computer ser. Berlin–Heidelberg: Springer. 6461:131–138.
8. Kirilenko, I., S. Grigorev, and D. Avdiukhin. 2013. Razrabotka sintaksicheskikh analizatorov v proektakh po avtomatizirovannomu rezhiniringu informatsionnykh sistem [Syntax analyzers development in automated reengineering of informational systems]. *Nauchno-tehnicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie* [St. Petersburg State Polytechnical University J. Computer Science. Telecommunications and Control Systems] 3(174):94–98.
9. Grigorev, S., E. Verbitskaia, A. Ivanov, M. Polubelova, and E. Mavchun. 2014. String embedded language support in integrated development environment. *10th Central and Eastern European Software Engineering Conference in Russia Proceedings*. Moscow. 21:1–21:11.
10. Yu, F., M. Alkhafaf, T. Bultan, and O. H. Ibarra. 2014. Automata-based symbolic string analysis for vulnerability detection. *Form. Method. Syst. Des.* 44(1):44–70.
11. Hanneforth, T. 2008. Finite-state machines: Theory and applications. Unweighted finite-state automata. 99 p. http://tagh.de/tom/wp-content/uploads/fsm_unweightedautomata.pdf (accessed April 22, 2016).
12. Hooimeijer, P., and M. Veane. 2011. An evaluation of automata algorithms for string analysis. *12th Conference (International) on Verification, Model Checking, and Abstract Interpretation Proceedings*. Berlin–Heidelberg: Springer-Verlag. 248–262.

Received February 26, 2016

Contributors

Polubelova Marina I. (b. 1993) — student, Saint Petersburg State University, 7/9 Universitetskaya Nab., St. Petersburg 199034, Russian Federation; polubelovam@gmail.com

Grigorev Semen V. (b. 1989) — PhD student, Saint Petersburg State University, 7/9 Universitetskaya Nab., St. Petersburg 199034, Russian Federation; Semen.Grigorev@jetbrains.com

НОРМАЛЬНЫЕ ФИЛЬТРЫ ПУГАЧЁВА ДЛЯ АВТОКОРРЕЛИРОВАННЫХ СТОХАСТИЧЕСКИХ СИСТЕМ, ЛИНЕЙНЫХ ОТНОСИТЕЛЬНО СОСТОЯНИЯ

И. Н. Синицын¹, Э. Р. Корепанов²

Аннотация: Рассматривается теория аналитического синтеза непрерывных условно-оптимальных фильтров Пугачёва для обработки процессов в гауссовых стохастических системах (СтС), линейных относительно вектора состояния. Для гауссовых систем первые работы по фильтрации были выполнены Липцером и Ширяевым, а для негауссовых — Пугачёвым и Синицыным. Приводятся алгоритмы нормальных фильтров для систем с некоррелированными и с автокоррелированными помехами. Приведен тестовый пример. Полученные алгоритмы положены в основу программного обеспечение (StS-Filter, 2016). Приведены некоторые обобщения.

Ключевые слова: автокоррелированная СтС; дифференциальная СтС; метод нормальной аппроксимации (МНА) апостериорной плотности; метод статистической линеаризации (МСЛ); нормальный фильтр Пугачёва (НФП); стохастическая система (СтС); СтС, линейная относительно состояния; условия Липцера–Ширяева; фильтр Липцера–Ширяева (ФЛШ)

DOI: 10.14357/08696527160204

1 Введение

Обработка процессов в ряде современных задач статистических научных исследований основана на использовании теории нелинейной фильтрации в СтС, линейных относительно состояния [1–5]. Для гауссовых СтС первые работы в этом направлении выполнены Липцером и Ширяевым [1], а для негауссовых СтС на основе нелинейной субоптимальной фильтрации — Пугачёвым и Синицыным (см., например, [3]). В [4, 5] рассмотрены вопросы синтеза алгоритмов нормальных условно-оптимальных фильтров и экстраполаторов Пугачёва (НФЭП) для обработки процессов в непрерывных и дискретных СтС, линейных относительно состояния, на основе методов нормальной аппроксимации (МНА) и статистической линеаризации (МСЛ). Особое внимание удалено синтезу НФЭП для СтС при условиях Липцера–Ширяева на базе аппроксимации нормальным субоптимальным квазилинейным фильтром, основанном на

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sinitsin@dol.ru

²Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ekogerepanov@piran.ru

МСЛ для нелинейных функций, зависящих от наблюдений. Для СтС высокой размерности путем выбора структурных функций, отражающих аналитическую природу наблюдаемой системы, синтезированы НФЭП, простые в компьютерной реализации и предназначенные для работы в режиме реального времени. Алгоритмы положены в основу модуля инструментального программного обеспечения «StS-Filter, 2015».

Настоящая статья посвящена вопросам синтеза НФП для обработки процессов в непрерывных СтС, линейных относительно состояния. В разд. 2 и 3 приводятся алгоритмы нормальных фильтров для СтС с некоррелированными помехами, а в разд. 2 и 4 — для СтС с автокоррелированными помехами. Раздел 5 содержит тестовый пример. Полученные алгоритмы положены в основу модуля программного обеспечения «StS-Filter, 2016».

2 Автокоррелированные гауссовские стохастические системы, линейные относительно состояния

Рассмотрим сначала нелинейную непрерывную (дифференциальную) СтС с некоррелированными гауссовскими помехами [3]:

$$\dot{X}_t = \varphi + \psi V_0; \quad X_{t_0} = X_0; \quad \dot{Y}_t = \varphi_1 + \psi_1 V_0, \quad Y_{t_0} = Y_0. \quad (1)$$

Здесь X_t и Y_t — векторы состояния и наблюдения размерности n_x и n_y ; $V_0 = \dot{W}_0$, где W_0 — помеха — векторный гауссовский стохастический процесс (СтП), обладающий нулевым математическим ожиданием и матрицей интенсивностей $\nu = \nu_0^v$; $\varphi = \varphi(X_t, Y_t, t)$; $\psi = \psi(X_t, Y_t, t)$, $\varphi_1 = \varphi_1(X_t, Y_t, t)$ и $\psi_1 = \psi_1(X_t, Y_t, t)$ — известные функции размерности $(n_x \times 1)$, $(n_x \times n_v)$, $(n_y \times 1)$ и $(n_y \times n_v)$ и удовлетворяющие следующим условиям Липпера–Ширяева [3]:

- функции φ и φ_1 линейны относительно состояния X_t , а функции ψ и ψ_1 не зависят от состояния X_t :

$$\left. \begin{array}{l} \varphi(X_t, Y_t, t) = a_1(Y_t, t)X_t + a_0(Y_t, t); \\ \varphi_1(X_t, Y_t, t) = b_1(Y_t, t)X_t + b_0(Y_t, t); \\ \psi(X_t, Y_t, t) = \bar{\psi}(Y_t, t); \\ \psi_1(X_t, Y_t, t) = \bar{\psi}_1(Y_t, t). \end{array} \right\} \quad (2)$$

Уравнения СтС (1), (2) понимаются в смысле Ито и имеют решение в среднем квадратическом (с.к.) [3].

Систему (1), (2) будем называть гауссовской, если X_0 и Y_0 — гауссовские случайные величины.

Важный частный случай (1), (2) составляют уравнения с аддитивными помехами, когда

$$\bar{\psi}(Y_t, t) = \psi_0(t); \quad \bar{\psi}_1(Y_t, t) = \psi_{10}(t). \quad (3)$$

Как известно [3], обычно состояние системы и помеха определяются различными, независимыми одно от другого стохастическими дифференциальными уравнениями (уравнением состояния системы и уравнением формирующего фильтра помехи). Однако для общности целесообразно при построении теории рассматривать случай, когда состояние X_t и помеха N_t определяются совместными стохастическими дифференциальными уравнениями. Таким образом, для задачи фильтрации при наличии автокоррелированной помехи в наблюдениях будем записывать уравнения наблюдения, системы и формирующего фильтра помехи в виде:

$$\dot{\hat{X}}_t = \varphi(X_t, Y_t, N_t, t) + \psi(X_t, Y_t, N_t, t) V_0; \quad (4)$$

$$\dot{Y}_t = \varphi_1(X_t, Y_t, N_t, t); \quad (5)$$

$$\dot{N}_t = \varphi_0(X_t, Y_t, N_t, t) + \psi_0(X_t, Y_t, N_t, t) V_0. \quad (6)$$

Условия Липцера–Ширяева (2) для рассматриваемых автокоррелированных СтС будут получены в разд. 4.

3 Нормальные фильтры для стохастических систем с некоррелированными гауссовскими помехами

Для гауссовой СтС (1), (2) в условиях невырожденности диффузационной матрицы $\sigma_1 = \sigma_1(Y_t, t) = \psi_1 \nu_0 \psi_1^T(Y_t, t)$ известны следующие точные уравнения нелинейной фильтрации по критерию минимума с.к. ошибки [3]:

$$\begin{aligned} \dot{\hat{X}}_t &= \left[a_1(Y_t, t) \hat{X}_t + a_0(Y_t, t) \right] + \left[R_t b_1(Y_t, t)^T + \left(\bar{\psi} \nu_0 \bar{\psi}_1^T \right) (Y_t, t) \right] \times \\ &\quad \times \left(\bar{\psi}_1 \nu_0 \bar{\psi}_1^T \right)^{-1} (Y_t, t) \left\{ \dot{Y}_t - \left[b_1(Y_t, t) \hat{X}_t + b_0(Y_t, t) \right] \right\}, \quad \hat{X}_{t_0} = \hat{X}_0; \end{aligned} \quad (7)$$

$$\begin{aligned} \dot{R}_t &= a_1(Y_t, t) R_t + R_t a_1(Y_t, t)^T + \left(\bar{\psi} \nu_0 \bar{\psi}^T \right) (Y_t, t) - \left[R_t b_1(Y_t, t)^T + \right. \\ &\quad \left. + \left(\bar{\psi}_1 \nu_0 \bar{\psi}_1^T \right) (Y_t, t) \right] \left(\bar{\psi}_1 \nu_0 \bar{\psi}_1^T \right)^{-1} (Y_t, t) \left[b_1(Y_t, t) R_t + \left(\bar{\psi}_1 \nu_0 \bar{\psi}^T \right) (Y_t, t) \right], \\ &\quad R_{t_0} = R_0, \end{aligned} \quad (8)$$

где \hat{X}_t — с.к. оценка СтП X_t ; R_t — ковариационная матрица ошибки фильтрации, равной $X_t - \hat{X}_t$.

Отметим, что, как и в случае линейной фильтрации при аддитивных шумах [3], уравнения дифференциального фильтра Липцера–Ширяева (ФЛШ) представляют собой замкнутую систему уравнений, определяющую \hat{X}_t и R_t . Поэтому с.к. оптимальную оценку СтП \hat{X}_t вектора состояния системы X_t и его

апостериорную ковариационную матрицу R_t , характеризующую точность с.к. оптимальной оценки СтП \hat{X}_t , можно вычислять по мере получения результатов наблюдений совместным интегрированием уравнений (7) и (8). В противоположность линейной фильтрации для ФЛШ нельзя вычислить R_t заранее (до получения результатов наблюдений), так как от последних зависят коэффициенты уравнения (8). Поэтому ФЛШ в данном случае должен выполнять интегрирование обоих уравнений (7) и (8). Это приводит к существенному повышению порядка оптимального фильтра. Если линейный фильтр всегда описывается уравнениями порядка n_x , то в рассматриваемом более общем случае с.к. оптимальный фильтр описывается уравнениями следующего порядка

$$Q_{\text{ФЛШ}} = n_x + \frac{n_x(n_x + 1)}{2} = \frac{n_x(n_x + 3)}{2}.$$

Очевидно, что ФЛШ будет совпадать с обобщенным (приближенным) фильтром Калмана–Бьюси, фильтрами второго порядка, гауссовыми фильтрами [2, 3].

В [4, 5] показано, что для гауссовой СтС (1)–(4) субоптимальные НФ на основе МНА одномерной апостериорной плотности и ФЛШ совпадают.

Будем искать условно-оптимальный фильтр Пугачёва (ФП) для оценки СтП \hat{X}_t в виде следующего уравнения [3–5]:

$$d\hat{X}_t = \alpha_t \xi(\hat{X}_t, Y_t, t) dt + \beta_t \eta(\hat{X}_t, Y_t, t) dY_t + \gamma_t dt. \quad (9)$$

Здесь $\xi = \xi(\hat{X}_t, Y_t, t)$ и $\eta = \eta(\hat{X}_t, Y_t, t)$ — некоторые известные структурные функции текущих значений оценки наблюдаемого СтП Y_t , \hat{X}_t и времени t ; α_t , β_t и γ_t — неизвестные функции времени. Если бы коэффициенты α_t , β_t и γ_t в (9) были известными функциями времени, то уравнение (9) определило бы фильтр того же порядка n_x , что и первое уравнение (1). Поэтому, естественно, возникает мысль попытаться непосредственно определить коэффициенты α_t , β_t и γ_t в уравнении (9) как функции времени из условия минимума с.к. ошибки при всех $t > t_0$. Это приводит к теории условно-оптимального ФП, когда уравнение ФП задается заранее, а оптимизируются только коэффициенты этого уравнения.

Итак, приходим к идее нахождения оптимального фильтра в некотором классе допустимых фильтров, определяемом условием, что поведение фильтра описывается дифференциальным уравнением заданного порядка и заданной формы.

Таким образом, мы отказываемся от абсолютной оптимизации и ограничиваемся условной оптимизацией в заданном ограниченном классе фильтров. Определив класс допустимых ФП, следует решить вопрос о том, какой фильтр в этом классе считается оптимальным. Следуя Пугачёву [3], будем считать оптимальным такой фильтр, который дает в известном смысле наилучшую оценку

при всех $t > t_0$. Иными словами, задача оптимизации фильтра при всех $t > t_0$ является задачей многокритериальной оптимизации. Такие задачи, как правило, не имеют решения.

Фильтр Калмана–Бьюси, дающий оптимальную линейную оценку состояния линейной системы в каждый момент $t > t_0$, является исключением [3]. Значит, надо определить такую оптимальность фильтра, при которой возможно решение задачи. Будем считать условно-оптимальным такой допустимый фильтр, который на каждом бесконечно малом интервале времени совершает оптимальный переход из того состояния, в котором он был в начале этого интервала, в новое состояние. Такой допустимый фильтр будем называть условно-оптимальным, или ФП. Тогда задача фильтрации сводится к нахождению оптимальных значений α_t , β_t и γ_t в любой момент $t \geq t_0$, обеспечивающих минимум с.к. ошибки фильтрации в бесконечно близкий будущий момент $s > t$, $s \rightarrow t$.

В [3] доказано, что ФП обладает тем свойством, что в данном классе допустимых фильтров не существует фильтра, который при данном начальном распределении СтП Y_t , X_t и \hat{X}_t в момент t_0 был бы лучше условно-оптимального при всех $t > t_0$. Это значит, что в терминологии теории многокритериальной оптимизации ФП представляет собой один из множества допустимых фильтров — оптимальный по Парето [5]. Общая теория ФП по с.к. критерию развита для уравнений общего вида и подробно изложена в [3].

Теория ФП обладает двумя несомненными преимуществами по сравнению с методами субоптимальной фильтрации [3].

Во-первых, она позволяет получать фильтры более низкого порядка и, следовательно, более простые в реализации.

Во-вторых, она дает возможность получать фильтры не меньшей, а при желании даже большей точности, чем фильтры, даваемые методами субоптимальной нелинейной фильтрации.

Применяя теорию ФП [3] к нормальным процессам в гауссовской СтС (1), (2), придем к нормальному ФП вида (9). Входящие в (9) коэффициенты α_t , β_t и γ_t определяются следующими уравнениями:

$$\left. \begin{aligned} \alpha_t m_1 + \beta_t m_2 + \gamma_t &= m_0; \\ m_0 = M^N[\varphi], \quad m_1 = M^N[\xi], \quad m_2 = M^N[\eta]; \\ \beta_t &= \kappa_{02} \kappa_{22}^{-1}; \\ \kappa_{02} = M^N \left[\left(X_t - \hat{X}_t \right) (a_1 X_t + a_0)^T \eta^T \right] + M^N \left[\bar{\psi} \nu_0 \bar{\psi}_1^T \eta^T \right]; \end{aligned} \right\} \quad (10)$$

$$\kappa_{22} = M^N \left[\eta \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \eta^T \right]; \quad (11)$$

$$\alpha_t \kappa_{11} + M^N \left[\left(\hat{X}_t - X_t \right) \left(\xi^T \alpha_t^T + \gamma_t^T \right) \frac{\partial \xi^T}{\partial \hat{X}_t} \right] = \kappa'_{01} - \beta_t \kappa'_{21}; \quad (12)$$

$$\left. \begin{aligned}
 \kappa'_{21} &= \mathbb{M}^N \{ [\eta(b_1 X_t + b_0) - m_2] \xi^T \}; \\
 \kappa'_{01} &= \kappa_{01} + \mathbb{M}^N \left[\left(X_t - \hat{X}_t \right) \frac{\partial \xi^T}{\partial t} \right] + \mathbb{M}^N \left\{ \left(X_t - \hat{X}_t \right) (b_1 X_t + b_0)^T + \right. \\
 &\quad \left. + \bar{\psi} \nu_0 \bar{\psi}_1^T - \beta_t \eta \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \right\} \left(\frac{\partial}{\partial Y_t} + \eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \right) \xi^T + \\
 &\quad + \frac{1}{2} \mathbb{M}^N \left[\left(X_t - \hat{X}_t \right) \left\{ \text{tr} \left[\bar{\psi}_1 \nu_0 \bar{\psi}_1^T \left(\frac{\partial}{\partial Y_t} + 2\eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \right) \frac{\partial^T}{\partial Y_t} \right] + \right. \right. \\
 &\quad \left. \left. + \text{tr} \left[\beta_t \eta \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \frac{\partial^T}{\partial \hat{X}_t} \right] \right\} \xi^T \right]; \\
 \kappa_{11} &= \mathbb{M}^N \{ [\xi - m_1] \xi^T \}; \\
 \kappa_{21} &= \mathbb{M}^N \{ [b_1 X_t + b_0 - m_2] \xi^T \}; \\
 \kappa_{01} &= \mathbb{M}^N \{ [a_1 X_t + a_0 - m_0] \xi^T \}.
 \end{aligned} \right\} \quad (13)$$

Здесь \mathbb{M}^N — символ математического ожидания для нормального распределения.

Точность НФП определяется уравнением:

$$\dot{R}_t = \mathbb{M}^N \left[(X_t - \hat{X}_t) (a_1 X_t + a_0)^T + (a_1 X_t + a_0) (X_t^T - \hat{X}_t^T) - \right. \\
 \left. - \beta_t \eta (Y_t, \hat{X}_t, t) \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \eta (Y_t, \hat{X}_t, t)^T \beta_t^T + \bar{\psi} \nu_0 \bar{\psi} \right], \quad R_{t_0} = R_0. \quad (14)$$

Таким образом, справедливо следующее утверждение [4, 5].

Теорема 1. Пусть для гауссовой системы (1), (2) выполнены условия невырожденности матрицы κ_{22} (11) и конечности величин κ_{ij} ($i, j = 0, 1, 2$). Тогда алгоритм НФП определяется уравнением (9), а коэффициенты α_t , β_t и γ_t — (10)–(14).

Теория НФП не позволяет получить нормальные с.к. оптимальные фильтры. Можно получить только ФП, которые в общем случае хуже с.к. оптимальных, но зато легко реализуемы. Однако если с.к. оптимальная оценка СтП \hat{X}_t вектора X_t удовлетворяет уравнению допустимого фильтра (9) при каких-либо коэффициентах времени α_t , β_t и γ_t , то уравнения теоремы 1, конечно, определяют именно эти α_t , β_t и γ_t и НФП будет с.к. оптимальным (последний в данном случае будет допустимым и, следовательно, оптимальным в классе допустимых фильтров). Кроме того, теория НФП дает возможность оценивать не все компоненты вектора состояния системы (в общем случае расширенного), а только некоторые из них. Для этого достаточно взять структурные функции ξ и η в (9), зависящими лишь от соответствующих компонент вектора \hat{X}_t . К примеру, взяв ξ и η в (18) (см. разд. 4) зависящими лишь от Y_t , t и оценок неизвестных

параметров системы, можно оценивать только параметры системы, не оценивая ее состояния. В таких случаях будут получаться НФП, порядок которых меньше размерности n_x расширенного вектора состояния.

Особое практическое значение имеет случай (1)–(3) с аддитивными (в общем случае негауссовскими) шумами. Следуя [3], проведем статистическую линеаризацию нелинейных (относительно Y_t) функций:

$$\begin{aligned} a_1(Y_t, t) X_t &\approx (k_{0x}^{a_1x} - k_{1x}^{a_1x}) m_t^x + \left(k_{0y}^{a_1x} - k_{1y}^{a_1x} \right) m_t^y + k_x^{a_1y} X_t + k_{0y}^{a_1x} Y_t; \\ b_1(Y_t, t) X_t &\approx \left(k_{0x}^{b_1x} - k_{1x}^{b_1x} \right) m_t^x + \left(k_{0y}^{b_1x} - k_{1y}^{b_1x} \right) m_t^y + k_x^{b_1y} X_t + k_{0y}^{b_1x} Y_t; \\ a_0(Y_t, t) &\approx \left(k_{0y}^{a_0} - k_{1y}^{a_0} \right) m_t^y + k_{0y}^{a_0} Y_t, \quad b_0(Y_t, t) \approx \left(k_{0y}^{b_0} - k_{1y}^{b_0} \right) m_t^y + k_{0y}^{b_0} Y_t. \end{aligned}$$

Тогда (1)–(5) приводятся к эквивалентной гауссовой системе — линейной относительно X_t^0, Y_t^0 и нелинейной относительно m_t^x, m_t^y :

$$\dot{X}_t = \bar{a} Y_t + \bar{a}_1 X_t + \bar{a}_0 + \bar{\psi} V; \quad \dot{Y}_t = \bar{b} Y_t + \bar{b}_1 X_t + \bar{b}_0 + \bar{\psi}_1 V. \quad (15)$$

Здесь введены обозначения:

$$\begin{aligned} \bar{a} &= k_y^{a_1x} + k_y^{a_0}; \quad \bar{a}_1 = k_x^{a_1x}; \quad a_0 = \left(k_{0y}^{a_0} - k_{1y}^{a_0} \right) m_t^y; \\ \bar{b} &= k_y^{b_1x}; \quad \bar{b}_1 = k_x^{b_1x}; \quad \bar{b}_0 = (k_{0y}^{b_0} - k_{1y}^{b_0}) m_t^y. \end{aligned}$$

Правые части уравнений (15) зависят от вероятностных моментов первого и второго порядка и определяются из следующей линейной дифференциальной системы для составного вектора $Z_t = [X_t^T Y_t^T]^T$:

$$\dot{m}_t^z = c m_t^z + c_0; \quad \dot{K}_t^z = c K_t^z + K_t^z c^T + l \nu l^T, \quad (16)$$

где

$$c = \begin{bmatrix} \bar{a}_1 & \bar{a} \\ \bar{b}_1 & \bar{b} \end{bmatrix}; \quad c_0 = \begin{bmatrix} \bar{a}_0 \\ \bar{b}_0 \end{bmatrix}; \quad l = \begin{bmatrix} \bar{\psi}_t \\ \bar{\psi}_{1t} \end{bmatrix}.$$

Используя теорию квазилинейной фильтрации [3], получим следующие уравнения квазилинейного субоптимального НФП:

$$\left. \begin{aligned} \dot{\hat{X}}_t &= \bar{a} Y_t + \bar{a}_1 \hat{X}_t + \bar{a}_0 + \beta_t \left[\dot{Y}_t - (\bar{b} Y_t + \bar{b}_1 \hat{X}_t + \bar{b}_0) \right]; \\ \beta_t &= R_t \bar{b}_1^T + (\bar{\psi} \nu \bar{\psi}_1^T) (\bar{\psi}_1 \nu \bar{\psi}_1^T)^{-1}; \\ \dot{R}_t &= \bar{a}_1 R_t + R_t \bar{a}_1^T + \bar{\psi} \nu \bar{\psi}^T - (R_t \bar{b}_1^T + \bar{\psi} \nu \bar{\psi}_1^T) (\bar{\psi}_1 \nu \bar{\psi}_1^T)^{-1} (\bar{b} R_t + \bar{\psi}_1 \nu \bar{\psi}_1^T). \end{aligned} \right\} (17)$$

Таким образом, имеем следующий результат [4, 5].

Теорема 2. Пусть уравнения гауссовой СмС (1)–(3) с аддитивными помехами допускают применение МСЛ. Тогда уравнения алгоритма квазилинейного НФП имеют вид (13).

4 Нормальные фильтры для стохастической системы с автокоррелированными помехами

В [3] показано, что задачи фильтрации и экстраполяции приводятся к случаю белого шума в наблюдениях преобразованием наблюдаемого сигнала системой, обратной формирующему фильтру, или дифференцированием уравнения наблюдения. В соответствии с этими результатами и теорией условно-оптимальной фильтрации будем вводить в уравнения допустимых фильтров производные наблюдаемого сигнала до s -го порядка включительно, если его s -я производная содержит белый шум. При этом производные, не содержащие белого шума, могут входить в уравнения фильтров нелинейно. В этом случае задачи условно-оптимальной фильтрации удается решать при произвольных уравнениях наблюдения и формирующего фильтра помехи, не требуя их линейности, как это приходится делать в теории оптимальной линейной фильтрации.

На основании сказанного определим класс допустимых фильтров в случае автокоррелированной помехи в наблюдениях уравнением:

$$\dot{\hat{X}}_t = \alpha_t \xi \left(Y_t, \dot{Y}_t, \dots, Y_t^{(s)}, \hat{X}_t, t \right) + \beta_t \eta \left(Y_t, \dot{Y}_t, \dots, Y_t^{(s)}, \hat{X}_t, t \right) \dot{Y}_t^{(s)} + \gamma_t. \quad (18)$$

Отметим, что можно и не вводить производные наблюдаемого сигнала в уравнения допустимых фильтров или вводить производные до порядка $h < s$. Для этого достаточно взять функцию ξ независимой от $Y_t^{(h+1)}, \dots, Y_t^{(s)}$ и положить $\eta \equiv 0$. Однако качество фильтрации значительно улучшается при вводе производных наблюдаемого сигнала до порядка s включительно.

Структурные функции ξ и η в (18) можно взять произвольно. Однако целесообразно их выбирать, приведя сначала задачу к случаю белого шума в наблюдениях путем s -кратного дифференцирования уравнения наблюдения и исключения некоторых компонент помехи с помощью уравнения наблюдения и уравнений, полученных из него ($s - 1$)-кратным дифференцированием.

При решении задач условно-оптимальной фильтрации для автокоррелированной помехи в наблюдениях ограничимся случаем гауссовского белого шума в уравнениях (4)–(6). Для приведения задачи фильтрации к случаю белого шума в наблюдениях будем дифференцировать уравнения наблюдения по формуле Ито [3] до появления белого шума в правой части.

Дифференцируя уравнение (4) по формуле Ито, получаем, опуская для краткости аргументы функций,

$$\begin{aligned} \frac{d}{dt} \dot{Y}_t = & \left(\varphi_{1t} + \varphi_{1y}^T \varphi_1 + \varphi_{1x}^T \varphi + \varphi_{1n}^T \varphi_0 + \frac{1}{2} \Phi_1 : \Psi_n \nu_0 \Psi^T \right) + \\ & + \left(\varphi_{1x}^T \psi_1 + \varphi_{1n}^T \psi_0 \right) V_0, \end{aligned}$$

где, как обычно, $\varphi_{1t} = \partial \varphi_1 / \partial t$, $\varphi_{1y} = (\partial / \partial Y_t) \varphi_1^T$, $\varphi_{1x} = (\partial / \partial X_t) \varphi_1^T$, $\varphi_{1n} = (\partial / \partial N_t) \varphi_1^T$, матрица Ψ определяется формулой $\Psi = [0 \ \psi^T \ \psi_0^T]^T$, а $\Phi_1 :$

$\Psi\nu\Psi^T$ представляет собой матрицу-столбец ($n_y \times 1$), элементами которой служат следы произведений матриц вторых производных соответствующих компонент векторной функции φ_1 по всем компонентам векторов СтП Y_t , X_t , N_t и матрицы $\Psi\nu\Psi^T$. Если $\varphi_{1x}^T\psi + \varphi_{1n}^T\psi_0 \neq 0$, то $s = 1$ и полученное уравнение является стохастическим дифференциальным уравнением, определяющим СтП \dot{Y}_t . Если же $\varphi_{1x}^T\psi + \varphi_{1n}^T\psi_0 \equiv 0$, то получим обыкновенное дифференциальное уравнение. В этом случае дифференцирование надо продолжать.

Предположим, что отличное от нуля слагаемое в выражении для V_0 появляется только после s -кратного дифференцирования (4), которое не может содержать меньше n_y независимых компонент помехи, а это может быть только при $n_N \geq \geq sn_y$. Тогда, положив $Y_t^{(0)} = Y_t$, $Y_t^{(k+1)} = dY_t^{(k)}/dt$ ($k = 0, 1, \dots, s-1$), будем иметь:

$$Y_t^{(k+1)} = \varphi_{k+1} \left(X_t, Y_t^{(0)}, N_t, t \right) \quad (k = 0, 1, \dots, s-1), \quad (19)$$

где функции $\varphi_2, \dots, \varphi_s$ определяются рекуррентной формулой

$$\begin{aligned} \varphi_{k+1}(X_t, Y_t, N_t, t) &= \varphi_{kt}(X_t, Y_t, N_t, t) + \varphi_{ky}(X_t, Y_t, N_t, t)^T \varphi_1(X_t, Y_t, N_t, t) + \\ &+ \varphi_{kx}(X_t, Y_t, N_t, t)^T \varphi(X_t, Y_t, N_t, t) + \varphi_{kn}(X_t, Y_t, N_t, t)^T \varphi_0(X_t, Y_t, N_t, t) + \\ &+ \frac{1}{2} \left(\Phi_k : \Psi\nu_0\Psi^T \right) (X_t, Y_t, N_t, t) \quad (k = 1, \dots, s-1). \end{aligned}$$

В этой формуле $\Phi_k : \Psi\nu_0\Psi^T$ — $(n_y \times 1)$ -матрица, элементами которой служат следы произведений матриц вторых производных соответствующих компонент векторной функции φ_k по всем компонентам векторов Y_t , X_t , N_t и матрицы $\Psi\nu_0\Psi^T$. Дифференцирование последнего уравнения (19), соответствующего $k = s-1$, приводит к уравнению:

$$\frac{d}{dt} Y_t^{(s)} = \varphi_{s+1} \left(X_t, Y_t^{(0)}, N_t, t \right) + \psi_1 \left(X_t, Y_t^{(0)}, N_t, t \right) V_0, \quad (20)$$

где

$$\begin{aligned} \varphi_{s+1} &= \varphi_{s+1}(X_t, Y_t, N_t, t) = \\ &= \varphi_{st}(X_t, Y_t, N_t, t) + \varphi_{sy}(X_t, Y_t, N_t, t)^T \varphi_1(X_t, Y_t, N_t, t) + \\ &+ \varphi_{sx}(X_t, Y_t, N_t, t)^T \varphi(X_t, Y_t, N_t, t) + \varphi_{sn}(X_t, Y_t, N_t, t)^T \varphi_0(X_t, Y_t, N_t, t) + \\ &+ \frac{1}{2} \left(\Phi_s : \Psi\nu_0\Psi^T \right) (X_t, Y_t, N_t, t); \end{aligned}$$

$$\begin{aligned} \psi_1 &= \psi_1(X_t, Y_t, N_t, t) = \\ &= \varphi_{sx}(X_t, Y_t, N_t, t)^T \psi(X_t, Y_t, N_t, t) + \varphi_{sn}(X_t, Y_t, N_t, t)^T \psi_0(X_t, Y_t, N_t, t). \end{aligned}$$

Таким образом, в результате s -кратного дифференцирования уравнения наблюдения уравнение (6) заменяется уравнениями (19) и стохастическим уравнением (20). Эти уравнения можно переписать в виде:

$$\left. \begin{aligned} \frac{d}{dt} Y_t^{(k)} &= Y_t^{(k+1)} \quad (k = 0, 1, \dots, s-1), \\ \frac{d}{dt} Y_t^{(s)} &= \varphi_{s+1} \left(X_t, Y_t^{(0)}, N_t, t \right) + \psi_1 \left(X_t, Y_t^{(0)}, N_t, t \right) V_0. \end{aligned} \right\} \quad (21)$$

Уравнения (21) вместе с (5) и (6), где $Y_t = Y_t^{(0)}$, представляют собой систему уравнений вида (1), (2) с расширенным вектором состояния $X'_t = [X_t^T N_t^T]^T$ (вместо X_t) и расширенным наблюдаемым вектором $Y'_t = [Y_t^{(0)T}, Y_t^{(1)T}, \dots, Y_t^{(s)T}]^T$ (вместо Y_t). Порядок этой системы уравнений, как правило, можно понизить, исключив некоторые или все компоненты вектора помехи N_t с помощью уравнений (19).

В этом случае обобщенные условия Липцера–Ширяева (2) принимают следующий вид:

$$\left. \begin{aligned} \varphi_{s+1} &= a_{1,s+1} (Y'_t, t) X'_t + a_{0,s+1} (Y'_t, t); \quad \psi_1 = \bar{\psi}_1 (Y_t, t); \\ \varphi_1 &= a_{1,1} (Y'_t, t) X'_t + a_{1,0} (Y'_t); \\ \varphi_0 &= a_{1,0} (Y'_t, t) X'_t + a_{0,0} (Y'_t, t); \quad \psi_0 = \bar{\psi}_0 (Y'_t, t). \end{aligned} \right\} \quad (22)$$

Используя теоремы 1 и 2 для расширенного вектора состояния, придем к следующим результатам. Для этого введем сначала обозначения:

$$\left. \begin{aligned} m_0 &= M^N [\varphi(X'_t, Y_t, t)]; \quad m_1 = M^N \left[\xi \left(Y'_t, \hat{X}_t, t \right) \right]; \\ m_2 &= M^N \left[\eta \left(Y'_t, \hat{X}_t, t \right) \varphi_{s+1} (X'_t, Y'_t, t) \right]; \\ \kappa_{02} &= M^N \left[\left(X_t - \hat{X}_t \right) \varphi_{s+1} (X'_t, Y_t, t)^T + (\bar{\psi} \nu_0 \bar{\psi}_1^T) (X'_t, Y_t, t) \right]; \\ \kappa_{22} &= M^N \left[\eta \left(Y'_t, \hat{X}_t, t \right) (\bar{\psi}_1 \nu_0 \bar{\psi}_1^T) (X'_t, Y_t, t) \eta \left(Y'_t, \hat{X}_t, t \right)^T \right]; \\ \kappa_{01} &= M^N \left\{ [\varphi_t (X'_t, Y_t, t) - m_0] \xi \left(Y'_t, \hat{X}_t, t \right)^T \right\}; \\ \kappa_{11} &= M^N \left\{ \left[\xi \left(Y'_t, \hat{X}_t, t \right) - m_1 \right] \xi \left(Y'_t, \hat{X}_t, t \right)^T \right\}; \\ \kappa_{21} &= M^N \left\{ [\varphi_{s+1} (X'_t, Y_t, t) - m_2] \xi \left(Y'_t, \hat{X}_t, t \right)^T \right\}; \\ \kappa'_{21} &= M^N \left\{ \left[\eta \left(Y'_t, \hat{X}_t, t \right) \varphi_{s+1} (X'_t, Y_t, t) - m_2 \right] \xi \left(Y'_t, \hat{X}_t, t \right)^T \right\}; \end{aligned} \right\} \quad (23)$$

$$\begin{aligned}
 \kappa'_{01} = & \mathbf{M}^N \left[(\varphi - m_0) \xi^T \right] + \mathbf{M}^N \left[(X_t - \hat{X}_t) \frac{\partial \xi^T}{\partial t} \right] + \\
 & + \sum_{k=1}^{s-1} \mathbf{M}^N \left[(X_t - \hat{X}_t) Y_t^{(k+1)T} \frac{\partial \xi^T}{\partial Y_t^{(k)}} \right] + \mathbf{M}^N \left\{ (X_t - \hat{X}_t) \varphi_{s+1}^T + \right. \\
 & \quad \left. + \bar{\psi} \nu_0 \bar{\psi}_1^T - \beta_t \eta \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \right\} \left(\frac{\partial}{\partial Y_t^{(s)}} + \eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \right) \xi^T + \\
 & + \frac{1}{2} \mathbf{M}^N \left[(X_t - \hat{X}_t) \left\{ \text{tr} \left[\bar{\psi}_1 \nu_0 \bar{\psi}_1^T \left(\frac{\partial}{\partial Y_t^{(s)}} + 2\eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \right) \frac{\partial^T}{\partial Y_t^{(s)}} \right] + \right. \right. \\
 & \quad \left. \left. + \text{tr} \left[\beta_t \eta \bar{\psi}_1 \nu_0 \bar{\psi}_1^T \eta^T \beta_t^T \frac{\partial}{\partial \hat{X}_t} \frac{\partial^T}{\partial \hat{X}_t} \right] \right\} \xi^T \right]. \quad (24)
 \end{aligned}$$

Здесь все функции без указания аргументов представляют собой их значения в момент t , например $\xi = \xi(Y_t^{(0)}, Y_t^{(1)}, \dots, Y_t^{(s)}, \hat{X}_t, t)$.

В частности, из

$$\begin{aligned}
 \xi(Y_t^{(0)}, Y_t^{(1)}, \dots, Y_t^{(s)}, \hat{X}_t, t) &= [\xi(Y_t^{(0)T}, Y_t^{(1)T}, \dots, Y_t^{(s)T}, \hat{X}_t^T)]^T; \\
 \eta(Y_t^{(0)}, Y_t^{(1)}, \dots, Y_t^{(s)}, \hat{X}_t, t) &= I
 \end{aligned}$$

следует, что $\kappa'_{21} = \kappa_{21}$ и $\kappa'_{01} = \kappa_{01}$.

Между коэффициентами НФП имеет место следующее уравнение связи [3]:

$$\alpha_t \kappa_{11} + \mathbf{M}^N \left[(\hat{X}_t - X_t) (\xi_t^T \alpha_t^T + \gamma_t^T) \frac{\partial \xi^T}{\partial \hat{X}_t} \right] = \kappa'_{01} - \beta_t \kappa'_{21}.$$

Тогда (24) принимает вид:

$$\alpha_t = \kappa_{01} - \beta_t \kappa_{21} \kappa_{11}^{-1}.$$

Для определения математических ожиданий в (23) и (24) достаточно знать совместное нормальное одномерное распределение процессов $Y_t^{(0)}, Y_t^{(1)}, \dots, Y_t^{(s)}, X_t, N_t$ и \hat{X}_t .

При расчете параметров совместного нормального распределения можно применить методы статистического моделирования и аналитического моделирования на основе МНА (МСЛ) [3].

Формула для производной по времени ковариационной матрицы ошибки фильтра Пугачёва имеет вид:

$$\begin{aligned}\dot{R}_t = M^N \left[\left(X_t - \hat{X}_t \right) \varphi(X_t, t)^T + \varphi(X_t, t) \left(X_t^T - \hat{X}_t^T \right) + \bar{\psi} \nu_0 \bar{\psi}^T - \right. \\ \left. - \beta_t \eta(Y'_t, \hat{X}_t, t) \bar{\psi}_1 \nu_0(t) \psi \psi_1^T \eta(Y'_t, \hat{X}_t, t)^T \right].\end{aligned}$$

Таким образом, имеем следующую теорему.

Теорема 3. Пусть для гауссовой автокоррелированной СтС (4)–(6) выполнены условия (22), конечны одномерные моменты и интегралы (23)–(24). Тогда алгоритм НФП определяется уравнениями теоремы 1 для расширенных векторов состояния X'_t и Y'_t .

В основе квазилинейного НФП для СтС с аддитивными помехами (4)–(6) лежат уравнения теоремы 2, но для расширенных векторов состояния X'_t и Y'_t .

Таким образом, в основе **теоремы 4** для $Z'_t = [X'^T_t Y'^T_t]^T$ лежат следующие соотношения:

$$\begin{aligned}\dot{X}'_t &= \bar{a}' Y'_t + \bar{a}'_1 X'_t + \bar{a}'_0 + \bar{\psi}' V_0; \quad \dot{Y}'_t = \bar{b}' Y'_t + \bar{b}'_1 X'_t + \bar{b}'_0 + \bar{\psi}'_1 V_0; \\ \dot{m}'_t &= c' m'_t + c'_0; \quad \dot{K}'_t = c' K'_t + K'_t c'^T + l' \nu_0 l'^T; \\ \dot{\hat{X}}'_t &= \bar{a}'_t \hat{X}'_t + \bar{a}'_1 \hat{X}'_t + \bar{a}'_0 + \beta'_t \left[\dot{Y}'_t - (\bar{b}' Y'_t + \bar{b}'_1 \hat{X}'_t + \bar{b}'_0) \right]; \\ \beta'_t &= R'_t \bar{b}'_1^T + \left(\bar{\psi}' \nu_0 \bar{\psi}'^T \right) \left(\bar{\psi}'_1 \nu_0 \bar{\psi}'_1^T \right); \\ \dot{R}'_t &= \bar{a}'_1 R'_t + R'_t \bar{a}'_1^T + \bar{\psi}' \nu_0 \bar{\psi}'^T - \\ &\quad - \left(R'_t \bar{b}'_1^T + \bar{\psi}' \nu_0 \bar{\psi}'^T \right) \left(\bar{\psi}'_1 \nu_0 \bar{\psi}'_1^T \right)^{-1} \left(\bar{b}' R'_t + \bar{\psi}'_1 \nu_0 \bar{\psi}'_1^T \right).\end{aligned}$$

5 Тестовый пример

Следуя [3], рассмотрим ФП для оценивания неизвестного параметра θ в уравнении системы

$$\dot{X}_t = -\theta X_t + V_1$$

по результатам наблюдения состояния X_t системы с аддитивной помехой, представляющей собой независимую от X_t нормально распределенную стационарную случайную функцию N_t с ковариационной функцией $k_n(\tau) = D e^{-a|\tau|}$.

В данном случае напишем уравнения (4)–(6) в виде:

$$\dot{Y}_t = Z_t = X_t + N_t. \quad (25)$$

Здесь

$$\dot{X}_t = -X_t \Theta_t + V_1; \quad \dot{\Theta}_t = 0; \quad \dot{N}_t = -aN_t + V_2,$$

где V_1 и V_2 — независимые нормально распределенные белые шумы, причем интенсивность белого шума V_2 равна $\nu_2 = 2Da$.

Чтобы найти подходящий класс допустимых фильтров, приведем задачу формально к случаю белого шума в наблюдениях и применим МНА [4, 5]. Дифференцируя уравнение наблюдения, заменим его уравнением:

$$\dot{Z}_t = -X_t \Theta_t - aN_t + V_1 + V_2.$$

Это уравнение вместе с уравнениями, определяющими расширенный вектор состояния $X'_t = [X_t \Theta_t N_t]^T$, образует систему уравнений вида (1)–(4) с

$$\begin{aligned} \varphi(Z_t, X_t, \Theta_t, N_t, t) &= [-X_t \Theta_t 0 \ -aN_t]^T; \\ \varphi_1(Z_t, X_t, \Theta_t, N_t, t) &= -X_t \Theta_t - aN_t; \\ \psi(Z_t, X_t, \Theta_t, N_t, t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T; \quad \psi_1(Z_t, t) = [1 \ 1]; \quad \nu = \begin{bmatrix} \nu_1 & 0 \\ 0 & \nu_2 \end{bmatrix}, \end{aligned}$$

для которой справедливы уравнения теоремы 1. Уравнения для \hat{X}_t и $\hat{\Theta}_t$, полученные в [4, 5], имеют вид:

$$\begin{aligned} \dot{\hat{X}}_t &= -\hat{X}_t \hat{\Theta}_t - R_{12} - \\ &\quad - (\nu_1 + \nu_2)^{-1} (\hat{X}_t R_{12} + \hat{\Theta}_t R_{11} + aR_{13}) (\dot{Z}_t - \hat{X}_t \hat{\Theta}_t - R_{12} - a\hat{N}_t); \\ \dot{\hat{\Theta}}_t &= -(\nu_1 + \nu_2)^{-1} (\hat{X}_t R_{22} + \hat{\Theta}_t R_{12} + aR_{23}) (\dot{Z}_t - \hat{X}_t \hat{\Theta}_t - R_{12} - a\hat{N}_t), \end{aligned}$$

где R_{ij} ($i, j = 1, 2, 3$) — элементы ковариационной матрицы вектора $[X_t \Theta_t N_t]^T$.

Правая часть уравнения для $\hat{\Theta}_t$ представляет собой полином второй степени относительно $\hat{\Theta}_t$ и \dot{Z}_t , содержащий члены $\hat{\Theta}$, $\hat{\Theta}_t^2$, \dot{Z}_t и $\hat{\Theta}_t \dot{Z}_t$ с коэффициентами, зависящими от других неизвестных функций времени \hat{X}_t , \hat{N}_t , R_{12} , R_{22} и R_{23} . Согласно основной идеи теории условно-оптимальной фильтрации все эти величины следует включить в уравнения подлежащих определению коэффициентов. Тогда, добавив еще член с Z_t (очевидно, это не может ухудшить качество фильтрации), получим уравнение класса допустимых ФП:

$$\dot{\hat{\Theta}}_t = \alpha_1 \hat{\Theta}_t + \alpha_2 \hat{\Theta}_t^2 + \alpha_3 Z_t + (\beta_1 + \beta_2 \hat{\Theta}_t) \dot{Z}_t + \gamma_t.$$

В данном случае $\varphi_2(X_t, \Theta_t, N_t, t) = -X_t \Theta_t - aN_t = aX_t - X_t \Theta_t - aZ_t$ (при нахождении класса допустимых фильтров функция φ_2 играет роль функции φ_1 в уравнениях теории оптимальной фильтрации), что позволяет получить для оптимальных β_1 и β_2 уравнения:

$$\begin{aligned} (\nu_1 + \nu_2) (\beta_1 + m_{0001} \beta_2) &= \\ &= m_{0111} - m_{0120} + a(m_{1001} - m_{1010} + m_{0110} - m_{0101}) + \nu_1; \end{aligned}$$

$$(\nu_1 + \nu_2) (m_{0001} \beta_1 + m_{0002} \beta_2) = \\ = m_{0112} - m_{0121} + a(m_{1002} - m_{1011} + m_{0111} - m_{0102}) + \nu_1 m_{0001},$$

где $m_{ijkl} = M^N [Z_t^i X_t^j \Theta_t^k \hat{\Theta}_t^l]$. При этом уравнения для оптимальных $\alpha_1, \alpha_2, \alpha_3$ и γ_t имеют следующий вид:

$$(m_{0002} - m_{0001}^2) \alpha_1 + (m_{0003} - m_{0002} m_{0001}) \alpha_2 + (m_{1001} - m_{1000} m_{0001}) \alpha_3 = \\ = \beta_1 [m_{0111} + am_{0101} - (m_{0110} + am_{1000} - am_{0100}) m_{0001}] + \\ + \beta_2 [m_{0112} + am_{1002} - am_{0102} - (m_{0111} + am_{1001} - am_{0101}) m_{0001}];$$

$$(m_{0003} - m_{0001} m_{0002}) \alpha_1 + (3m_{0004} - 2m_{0013} - m_{0003}^2) \alpha_2 + \\ + (3m_{1002} - 2m_{1011} - m_{1000} m_{0002}) \alpha_3 = \beta_1 [m_{0112} + am_{1002} - am_{0102} - \\ - (m_{0110} + am_{1000} - am_{0100}) m_{0002}] + \beta_2 [3(m_{0113} + am_{1003} - am_{0103}) - \\ - 2(m_{0122} + am_{1012} - am_{0112}) + (2\nu_1 - m_{0111} - am_{1001} + am_{0101}) m_{0002}] + \\ + 2(\nu_1 + \nu_2) [\beta_1^2 (m_{0010} - m_{0001}) + \beta_1 \beta_2 (2m_{0011} - 3m_{0002}) + \\ + \beta_2^2 (m_{0012} - 2m_{0003})];$$

$$(m_{1001} - m_{1000} m_{0001}) \alpha_1 + (m_{1002} - m_{1000} m_{0002}) \alpha_2 + (m_{2000} - m_{1000}^2) \alpha_3 = \\ = \beta_1 [m_{1110} + am_{2000} - am_{1100} - (m_{0110} + am_{1000} - am_{0100}) m_{1000}] + \\ + \beta_2 [m_{1111} + am_{2001} - am_{1101} - (m_{0111} + am_{1001} - am_{0101}) m_{1000}].$$

Оптимальное значение γ_t определяется формулой:

$$\gamma_t = -\alpha_1 m_{0001} - \alpha_2 m_{0002} - \alpha_3 m_{1000} + \beta_1 (m_{0110} + am_{1000} - am_{0100}) + \\ + \beta_2 (m_{0111} + am_{1001} - am_{0101}).$$

При использовании [6] для синтеза алгоритмов НФП было разработано экспериментальное программное обеспечение на основе нормальных суб- и условно-оптимальных ФП для уравнения (25). Показано, что линейный НФП проще ФЛШ и обеспечивает погрешность 20%–30%. Учет членов второго порядка позволяет снизить ошибку до 10%, что достаточно, например, для применений в области безопасности технических систем.

6 Заключение

Разработана теория аналитического синтеза непрерывных НФП для обработки процессов в гауссовских системах, линейных относительно состояния, при некоррелированных и автокоррелированных помехах.

Алгоритмы положены в основу программного обеспечения «StS-Filter» (version 2016) для стохастических информационных технологий [6]. Приведен тестовый пример.

Результаты допускают обобщение на случай аналитического синтеза нормальных экстраполаторов и интерполяторов Пугачёва.

Важное прикладное значение имеют задачи обобщения результатов на случай непрерывных СтС с негауссовскими помехами и дискретных СтС на основе [7].

Литература

1. Липцер Р. И., Ширяев А. Н. Статистика случайных процессов. — М.: Наука, 1974. 476 с.
2. Ройтенберг Я. Н. Автоматическое управление. — 3-е изд., перераб. и доп. — М.: Наука, 1992. 576 с.
3. Синицын И. Н. Фильтры Калмана и Пугачева. — 2-е изд. — М.: Логос, 2007. 776 с.
4. Синицын И. Н., Корепанов Э. Р. Нормальные условно-оптимальные фильтры Пугачёва для дифференциальных стохастических систем, линейных относительно состояния // Информатика и ее применения, 2015. Т. 9. Вып. 2. С. 30–38.
5. Синицын И. Н., Корепанов Э. Р. Нормальные условно-оптимальные фильтры и экстраполаторы Пугачёва для стохастических систем, линейных относительно состояния // Информатика и её применения, 2016 (в печати). Т. 10. Вып. 2.
6. Корепанов Э. Р. Стохастические информационные технологии на основе фильтров Пугачёва // Информатика и её применения, 2011. Т. 5. Вып. 2. С. 36–57.
7. Синицын И. Н. Параметрическое статистическое и аналитическое моделирование распределений в нелинейных стохастических системах на многообразиях // Информатика и её применения, 2013. Т. 7. Вып. 2. С. 4–16.

Поступила в редакцию 25.02.16

NORMAL PUGACHEV FILTERS FOR STATE LINEAR AUTOCORRELATED STOCHASTIC SYSTEMS

I. N. Sinitsyn and E. R. Korepanov

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: The analytical synthesis theory of continuous conditionally optimal Pugachev filters for processing in linear state stochastic systems (StS) with uncorrelated and autocorrelated noises is presented. For non-Gaussian StS, first works belong to Pugachev and Sinitsyn. Basic algorithms for state linear StS with uncorrelated noises are given. Generalization of algorithms for autocorrelated state linear Sts is presented. A test example for the software tool «StS-Filter, 2016» is described in details. Some generalizations are mentioned.

Keywords: autocorrelated noise; Liptser–Shiryaev filter (LSF); Liptser–Shiryaev conditions; normal approximation method (NAM) for *a posteriori* density; normal conditionally optimal Pugachev filter (NPF); stochastic systems (StS); state linear StS; statistical linearization method (SLM)

DOI: 10.14357/08696527160204

References

1. Liptser, R. Sh., and A.N. Shiryaev. 1974. *Statistika sluchaynykh protsessov* [Statistics of stochastic processes]. Moscow: Nauka. 476 p.
2. Roytenberg, Ya. N. 1992. *Avtomatischeskoe upravlenie* [Automatic control]. 3rd ed. Moscow: Nauka. 576 p.
3. Sinitsyn, I. N. 2007. *Fil'try Kalmana i Pugacheva* [Kalman and Pugachev filters]. 2nd ed. Moscow: Logos. 776 p.
4. Sinitsyn, I. N., and E. R. Korepanov. 2015. Normal'nye uslovno-optimal'nye fil'try Pugacheva dlya differentsiyal'nykh stokhasticheskikh sistem, lineynykh otnositel'no sostoyaniya [Normal Pugachev filters for state linear stochastic systems]. *Informatika i ee Primeneniya — Inform. Appl.* 9(2):30–38.
5. Sinitsyn, I. N., and E. R. Korepanov. 2016 (in press). Normal'nye uslovno-optimal'nye fil'try i ekstrapolyatory Pugacheva dlya stokhasticheskikh sistem, lineynykh otnositel'no sostoyaniya [Normal conditionally optimal filters and extrapolators for state linear stochastic systems]. *Informatika i ee Primeneniya — Inform. Appl.* 10(2).
6. Korepanov, E. R. 2011. Stokhasticheskie informatsionnye tekhnologii na osnove fil'trov Pugacheva [Stochastic informational technologies based on Pugachev filters]. *Informatika i ee Primeneniya — Inform. Appl.* 5(2):36–57.
7. Sinitsyn, I. N. 2013. Parametricheskoe statisticheskoe i analiticheskoe modelirovanie raspredeleniy v nelineynykh stokhasticheskikh sistemakh na mnogoobraziyakh [Parametric statistical and analytical modeling of distributions in stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 7(2):4–16.

Received February 25, 2016

Contributors

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sinitsin@dol.ru

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ekorepanov@ipiran.ru

МОДИФИЦИРОВАННЫЕ ЭЛЛИПСОИДАЛЬНЫЕ СУБОПТИМАЛЬНЫЕ ФИЛЬТРЫ ДЛЯ НЕЛИНЕЙНЫХ СТОХАСТИЧЕСКИХ СИСТЕМ НА МНОГООБРАЗИЯХ*

И. Н. Синицын¹, В. И. Синицын², Э. Р. Корепанов³

Аннотация: Разработана теория аналитического синтеза по критерию минимума средней квадратической ошибки модифицированных эллипсоидальных субоптимальных фильтров (СОФ) для нелинейных дифференциальных стохастических систем (СтС) на гладких многообразиях (МСтС) на основе эллипсоидальной аппроксимации ненормированных апостериорных распределений. Рассмотрены случаи гауссовых и негауссовых МСтС. Алгоритмы МЭСОФ по сравнению с алгоритмами эллипсоидальных СОФ (ЭСОФ) обладают достаточной простотой при использовании символьных методов записи. Алгоритмы модифицированных ЭСОФ (МЭСОФ) положены в основу модуля инструментального программного обеспечения («StS-Filter», 2016).

Ключевые слова: винеровский шум; метод эллипсоидальной аппроксимации (МЭА); модифицированный ЭСОФ (МЭСОФ); ненормированная одномерная апостериорная характеристическая функция; пуассоновский шум; субоптимальный фильтр (СОФ); уравнения точности и чувствительности МЭСОФ; эллипсоидальный СОФ (ЭСОФ)

DOI: 10.14357/08696527160205

1 Введение

В [1] представлена теория СОФ на базе методов нормальной аппроксимации (МНА) и статистической линеаризации (МСЛ), а также метода ортогональных разложений (МОР) для МСтС с винеровскими шумами в уравнениях наблюдения и винеровскими и пуассоновскими шумами в уравнениях состояния. В основу СОФ были положены точные нелинейные уравнения для апостериорного одномерного распределения.

В [2] рассмотрено развитие на случай, когда апостериорное одномерное распределение ошибки фильтрации допускает эллипсоидальную аппроксимацию [3]. Получены точные фильтрационные уравнения, а также уравнения точности

*Работа выполнена при поддержке РФФИ (проект 15-07-02244).

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sinitsin@dol.ru

²Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vsinitsin@ipiran.ru

³Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ekogerepanov@ipiran.ru

и чувствительности на основе МОР, даны элементы эллипсоидального анализа распределений, выведены уравнения ЭСОФ по методам эллипсоидальной аппроксимации (МЭА) и эллипсоидальной линеаризации (МЭЛ).

Рассмотрим теорию аналитического синтеза МЭСОФ на основе приближенного решения по МЭА (МЭЛ) [2] линейного фильтрационного уравнения для ненормированной апостериорной характеристической функции в условиях [2].

2 Точные фильтрационные уравнения для ненормированных распределений

Будем рассматривать задачу фильтрации состояния систем, моделями которых могут служить стохастические дифференциальные уравнения, понимаемые в смысле Ито. При этом стохастические дифференциальные уравнения модели изучаемой системы могут иметь неизвестные параметры и, как правило, всегда содержат параметры, известные с ограниченной точностью. Поэтому возникает задача непрерывного оценивания неизвестных параметров системы (точнее, ее модели) по результатам непрерывных наблюдений. Предположим, что правые части уравнений зависят от конечного множества неизвестных параметров, которые будем рассматривать как компоненты вектора параметров θ . Одним из возможных подходов в таких случаях является следующий прием: неизвестный векторный параметр θ считают стохастическим процессом (СтП) $\Theta = \Theta_t$, который определяется дифференциальным уравнением $\dot{\Theta}_t = 0$, и включают компоненты этого векторного процесса в вектор состояния системы («расширяют» вектор состояния путем включения в него неизвестных параметров в качестве дополнительных компонент). Таким образом, задача непрерывного оценивания неизвестных параметров модели системы сводится к задаче непрерывного оценивания состояния системы с расширенным вектором состояния. От неизвестных параметров могут зависеть и уравнения наблюдения. Эти параметры следует включить в вектор Θ и, следовательно, в расширенный вектор состояния.

Пусть векторный СтП $[X_t^T Y_t^T]^T$ определяется системой векторных стохастических дифференциальных уравнений Ито [4]:

$$dX_t = \varphi(X_t, Y_t, \Theta, t) dt + \psi'(X_t, Y_t, \Theta, t) dW_0 + \\ + \int_{R_0^q} \psi''(X_t, Y_t, \Theta, t, v) P^0(dt, dv), \quad X(t_0) = X_0; \quad (1)$$

$$dY_t = \varphi_1(X_t, Y_t, \Theta, t) dt + \psi'_1(X_t, Y_t, \Theta, t) dW_0 + \\ + \int_{R_0^q} \psi''_1(X_t, Y_t, \Theta, t, v) P^0(dt, dv), \quad Y(t_0) = Y_0. \quad (2)$$

Здесь $Y_t = Y(t)$ — n_y -мерный наблюдаемый СтП, $Y_t \in \Delta^y$ (Δ^y — гладкое многообразие наблюдений); $X_t = X(t)$ — n_x -мерный ненаблюдаемый СтП (вектор состояния), $X_t \in \Delta^x$ (Δ^x — гладкое многообразие состояний); $W_0 = W_0(t)$ — n_w -мерный винеровский СтП ($n_w \geq n_y$) интенсивности $\nu_0 = \nu_0(\Theta, t)$; $P^0(\Delta, A) = P(\Delta, A) - \mu_P(\Delta, A)$, $P(\Delta, A)$ представляет собой для любого множества A простой пуассоновский СтП, а $\mu_P(\Delta, A)$ — его математическое ожидание, причем

$$\mu_P(\Delta, A) = MP(\Delta, A) = \int_{\Delta} \nu_P(\tau, A) d\tau;$$

$\nu_P(\Delta, A)$ — интенсивность соответствующего пуассоновского потока событий, $\Delta = (t_1, t_2]$; интегрирование по v распространяется на все пространство R^q с выколотым началом координат; Θ — вектор случайных параметров размерности n_Θ ; $\varphi = \varphi(X_t, Y_t, \Theta, t)$, $\varphi_1 = \varphi_1(X_t, Y_t, \Theta, t)$, $\psi' = \psi'(X_t, Y_t, \Theta, t)$ и $\psi'_1 = \psi'_1(X_t, Y_t, \Theta, t)$ — известные функции, отображающие $R^{n_x} \times R^{n_y} \times R$ соответственно в R^{n_x} , R^{n_y} , $R^{n_x n_w}$ и $R^{n_y n_w}$; $\psi'' = \psi''(X_t, Y_t, \Theta, t, v)$ и $\psi''_1 = \psi''_1(X_t, Y_t, \Theta, t, v)$ — известные функции, отображающие $R^{n_x} \times R^{n_y} \times R^q$ в R^{n_x} и R^{n_y} . Требуется найти оценку \hat{X}_t СтП X_t в каждый момент времени t по результатам наблюдения СтП $Y(\tau)$ до момента t , $Y_{t_0}^T = \{Y(\tau) : t_0 \leq \tau < t\}$.

Предположим, что выполнены условия

- уравнение состояния имеет вид (1);
- уравнение наблюдения (2), во-первых, не содержит пуассоновского шума ($\psi''_1 \equiv 0$), а во-вторых, коэффициент при винеровском шуме ψ'_1 в уравнениях наблюдения не зависит от состояния ($\psi'_1(X_t, Y_t, \Theta, t) = \psi'_1(Y_t, \Theta, t)$).

В этом случае уравнения задачи нелинейной фильтрации имеют следующий вид:

$$dX_t = \varphi(X_t, Y_t, \Theta, t) dt + \psi'(X_t, Y_t, \Theta, t) dW_0 + \int_{R_0^q} \psi''(X_t, Y_t, \Theta, t, v) P^0(dt, dv), \quad X(t_0) = X_0; \quad (3)$$

$$dY_t = \varphi_1(X_t, Y_t, \Theta, t) dt + \psi_1(Y_t, \Theta, t) dW_0, \quad Y(t_0) = Y_0. \quad (4)$$

Будем считать, что выполнены условия существования и единственности СтП, определяемого (3) и (4) при соответствующих начальных условиях [4, 5].

Как известно [5, 6], для любых СтП X_t и Y_t оптимальная оценка \hat{X}^T , минимизирующая средний квадрат ошибки в каждый момент времени t , представляет собой апостериорное математическое ожидание СтП X_t :

$$\hat{X}_t = M[X_t | Y_{t_0}^T].$$

Чтобы найти это условное математическое ожидание, необходимо знать $p_t = p_t(x)$ и $g_t = g_t(\lambda)$ — апостериорную одномерную плотность и характеристическую функцию распределения СТП X_t . Соответствующие точные уравнения линейной фильтрации приведены в [2].

Введем ненормированные одномерные апостериорные плотность $\tilde{p}_t(x, \Theta)$ и характеристическую функцию $\tilde{g}_t(\lambda, \Theta)$ согласно формулам:

$$\tilde{p}_t(x, \Theta) = \mu_t p_t(x, \Theta); \quad \tilde{g}_t(\lambda, \Theta) = M_{\Delta^x}^{p_t} \left[e^{i\lambda^T X_t} \mu_t \right] = \mu_t g_t(\lambda, \Theta).$$

Тогда, обобщая [6] на случай уравнений (3) и (4), получим следующее точное уравнение среднеквадратической (с.к.) оптимальной нелинейной фильтрации:

$$\begin{aligned} d\tilde{g}_t(\lambda, \Theta) &= M_{\Delta^x}^{\tilde{p}_t} \left\{ \left[i\lambda^T \varphi(X, Y_t, \Theta, t) - \frac{1}{2} (\psi' \nu_0 \psi'^T) (X, Y_t, \Theta, t) + \right. \right. \\ &\quad \left. \left. + \int_{R_0^q} \left[e^{i\lambda^T \psi''(X, Y_t, \Theta, t, v)} - 1 - i\lambda^T \psi''(X, Y_t, \Theta, t, v) \right] \nu_P(\Theta, t, dv) \right] e^{i\lambda^T X} \right\} dt + \\ &\quad + M_{\Delta^x}^{\tilde{p}_t} \left\{ \left[\varphi_1(X, Y_t, \Theta, t)^T + i\lambda^T (\psi' \nu_0 \psi'^T) (X, Y_t, \Theta, t) \right] e^{i\lambda^T X} \right\} \times \\ &\quad \times (\psi' \nu_0 \psi'^T)^{-1} (Y_t, \Theta, t) dY_t. \quad (5) \end{aligned}$$

Если, следуя [7], функция ψ'' в (3) допускает представление

$$\psi'' = \psi' \omega(\Theta, v),$$

где $P^0(\Delta, A) = P^0((0, t], dv)$, то уравнения (3) и (4) примут следующий вид:

$$\begin{cases} \dot{X}_t = \varphi(X_t, Y_t, \Theta, t) + \psi'(X_t, Y_t, \Theta, t) V(\Theta, t), \quad X(t_0) = X_0; \\ \dot{Y}_t = \varphi(X_t, Y_t, \Theta, t) + \psi_1(Y_t, \Theta, t) V_0(\Theta, t), \quad Y(t_0) = Y_0. \end{cases} \quad (6)$$

Здесь

$$\begin{aligned} V_0(\Theta, t) &= \dot{W}_0(\Theta, t); \quad V(\Theta, t) = \dot{W}(\Theta, t); \\ \bar{W}(\Theta, t) &= W_0(\Theta, t) + \int_{R_0^q} \omega(\Theta, v) P^0((0, t], dv), \end{aligned}$$

где $\nu_P(\Theta, t, v) dv = [\partial \mu(\Theta, t, v) / \partial t] dv$ — интенсивность пуассоновского потока скачков, равных $\omega(\Theta, t)$. При этом логарифмические производные от одномерных характеристических функций определяются известными формулами:

$$\begin{aligned}\chi^{W_0}(\rho; \Theta, t) &= -\frac{1}{2} \rho^T \nu_0(\Theta, t) \rho; \\ \chi^{\bar{W}}(\rho; \Theta, t) &= -\frac{1}{2} \rho^T(\Theta, t) \rho^T + \int_{R_0^q} \left[e^{i\rho^T \omega(\Theta, v)} - 1 - i\rho^T \omega(\Theta, v) \right] \nu_P(\Theta, t, v) dv.\end{aligned}$$

В таком случае интегральный член в (5) допускает следующую запись:

$$\begin{aligned}\gamma &= \\ &= \int_{R_0^q} \left[e^{i\lambda^T \psi''(X_t, Y_t, \Theta, t) \omega(\Theta, v)} - 1 - i\lambda^T \psi''(X_t, Y_t, \Theta, t) \omega(\Theta, v) \right] \nu_P(\Theta, t, v) dv.\end{aligned}\quad (7)$$

Очевидно, что для гауссовой МСтС $\gamma \equiv 0$. Тогда приходим к результатам [8].

Таким образом, можно сформулировать следующие утверждения разд. 2.

Теорема 2.1. *Пусть для негауссовой МСтС (3), (4) выполнены условия существования и единственности. Тогда уравнение с.к. оптимальной нелинейной фильтрации для ненормированной характеристической функции $\tilde{g}_t(\lambda, \Theta)$ имеет вид (5).*

Теорема 2.2 [7]. *Пусть для гауссовой МСтС (6) выполнены условия существования и единственности. Тогда уравнение с.к. оптимальной нелинейной фильтрации для ненормированной характеристической функции имеет вид (5) при условии (7).*

Необходимость обработки результатов наблюдений в реальном масштабе времени непосредственно в процессе эксперимента привела к появлению ряда приближенных методов оптимальной нелинейной фильтрации, называемых обычно методами субоптимальной фильтрации [6]. Одни приближенные методы основаны на приближенном решении фильтрационных уравнений, а другие — на превращении формул для стохастических дифференциалов оптимальной оценки \hat{X}_t и апостериорной ковариационной матрицы ошибки R_t в стохастические дифференциальные уравнения для \hat{X}_t и R_t путем разложения функций φ , φ_1 и ψ_1 или φ , φ_1 , ψ' , ψ'' , ψ и ψ_1 в степенные ряды и отбрасывания остаточных членов.

Для приближенного решения уравнения для апостериорной одномерной характеристической функции $g_1(\lambda, \Theta)$ вектора X_t можно использовать методы аналитического моделирования, основанные на параметризации одномерных распределений СтП, определяемого стохастическим дифференциальным уравнением [6]. Эти методы позволяют изучить стохастические дифференциальные уравнения для параметров апостериорного распределения. Простейшим таким методом является МНА апостериорного распределения. Исключительно важное практическое значение имеют квазилинейные фильтры, получаемые с помощью

методов эквивалентной линеаризации [6]. Эллипсоидальные СОФ, основанные на приближенном решении уравнений для нормированных распределений, рассмотрены в [2].

3 Основные результаты

3.1. Аппроксимируем ненормированную апостериорную плотность вероятности формулой:

$$\tilde{p}_t(x) \approx p_t^*(u) = w(u) \left[\mu + \sum_{\nu=1}^N c_\nu p_\nu(u) \right]. \quad (8)$$

Имеем

$$c_\nu = \mu_t M^{\mathcal{O}A} [q_\nu(U_t)] = \left[q_\nu(U_\lambda) \tilde{g}_t^{\mathcal{O}A}(\lambda) \right]_{\lambda=0},$$

причем

$$\begin{aligned} u &= (x^T - \hat{X}_t^T) C_t (x - \hat{X}_t); \quad U_t = (X_t^T - \hat{X}_t^T) C_t (X_t - \hat{X}_t); \\ U_\lambda &= \left(\frac{\partial^T}{i\partial\lambda} - \hat{X}_t \right) C_t \left(\frac{\partial}{i\partial\lambda} - \hat{X}_t \right). \end{aligned}$$

Чтобы составить стохастические дифференциальные уравнения для коэффициентов c_ν , надо найти стохастический дифференциал Ито произведения $q_\chi(u)\tilde{g}_t(\lambda)$, имея в виду, что u зависит от $\hat{X}_t = m_t/\mu_t$ и что m_t и μ_t определяются стохастическими дифференциальными уравнениями. Потом следует заменить переменные x и u операторами $\partial/i\partial\lambda$ и U_λ , выполнить дифференцирование и после этого положить $\lambda = 0$.

3.2. Уравнения для m_t и μ_t с функцией $\hat{\varphi}_1$, определяемой формулой

$$\hat{\varphi}_1 = M_{\Delta^x}^{p_t} [\varphi_1],$$

с учетом обозначений

$$\sigma_0 = \psi\nu_0\psi^T; \quad \sigma_1 = \psi\nu_0\psi_1^T; \quad \sigma_2 = \psi_1\nu_0\psi_1^T$$

можно представить в виде:

$$\begin{aligned} dm_t &= \left[\mu_t f_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N c_\nu f_\nu(Y_t, \hat{X}_t, \Theta, t) \right] dt + \\ &\quad + \left[\mu_t h_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N c_\nu h_\nu(Y_t, \hat{X}_t, \Theta, t) \right] dY_t; \quad (9) \end{aligned}$$

$$d\mu_t = \left[\mu_t b_0(Y_t, \hat{X}_t, t) + \sum_{\nu=1}^N c_\nu b_\nu(Y_t, \hat{X}_t, \Theta, t) \right] dY_t, \quad (10)$$

где

$$\left. \begin{aligned} f_0 &= f_0(Y_t, \hat{X}_t, \Theta, t) = M_{\Delta^x}^w[\varphi] ; \\ f_\nu &= f_\nu(Y_t, \hat{X}_t, \Theta, t) = M_{\Delta^x}^{wp_\nu}[\varphi] ; \\ h_0 &= h_0(Y_t, \hat{X}_t, \Theta, t) = \\ &= M_{\Delta^U}^w \left[\sigma_1(Y_t, \Theta, t) + X \varphi_1(X, Y_t, \Theta, t)^T \right] \sigma_2(Y_t, \Theta, t)^{-1} ; \\ h_\nu &= h_\nu(Y_t, \hat{X}_t, \Theta, t) = \\ &= M_{\Delta^U}^{wp_\nu} \left[\sigma_1(X, Y_t, \Theta, t) + X \varphi_1(X, Y_t, \Theta, t)^T \right] \sigma_2(Y_t, \Theta, t)^{-1} ; \\ b_0 &= b_0(Y_t, \hat{X}_t, \Theta, t) = M_{\Delta^U}^w \left[\varphi_1(X, Y_t, \Theta, t)^T \right] \sigma_2(Y_t, \Theta, t)^{-1} ; \\ b_\nu &= b_\nu(Y_t, \hat{X}_t, \Theta, t) = M_{\Delta^U}^{wp_\nu} [\varphi_1(X, Y_t, \Theta, t)] \sigma_2(Y_t, \Theta, t)^{-1} . \end{aligned} \right\} \quad (11)$$

3.3. Запишем уравнения (9) и уравнение для ненормированной характеристической функции в виде:

$$dm_t = f dt + h dY_t ; \quad d\mu_t = b dY_t ; \quad d\tilde{g}_t = Adt + B dY_t .$$

Здесь обозначено:

$$\begin{aligned} f &= \mu_t f_0 + \sum_{\nu=1}^N c_\nu f_\nu ; \quad h = \mu_t h_0 + \sum_{\nu=1}^N c_\nu h_\nu ; \quad b = \mu_t b_0 + \sum_{\nu=1}^N c_\nu b_\nu ; \\ A &= M_{\Delta^x}^{\tilde{p}_t} \left\{ i \lambda^T \varphi(X, Y_t, \Theta, t) - \frac{1}{2} \lambda^T \left(\psi' \nu_0 \psi'^T \right) (X, Y_t, \Theta, t) \lambda \times \right. \\ &\quad \times \int_{R_0^q} \left[e^{i \lambda^T \psi''(X, Y_t, \Theta, t, v)} - 1 - i \lambda^T \psi''(X, Y_t, \Theta, t, v) \right] \nu_P(t, dv) e^{i \lambda^T X} \Big\} ; \\ B &= M_{\Delta^x}^{\tilde{p}_t} \left[\varphi_1(X, Y_t, \Theta, t)^T + i \lambda^T \left(\psi' \nu_0 \psi'^T \right) (X, Y_t, \Theta, t) \right] \times \\ &\quad \times e^{i \lambda^T X} \left(\psi'_1 \nu_0 \psi'^T \right)^{-1} (X, Y_t, \Theta, t) . \end{aligned}$$

3.4. Вычислим дифференциал Ито полинома $q_\chi(u)$,

$$u = \left(x^T - \frac{m_t^T}{\mu_t} \right) C_t \left(x - \frac{m_t}{\mu_t} \right) .$$

Имея в виду, что $\hat{X}_t = m_t/\mu_t$, находим:

$$\begin{aligned}
 dq_\chi(u) = & \left\{ q'_\chi(u) \left(x^T - \hat{X}_t^T \right) \dot{C}_t \left(x - \hat{X}_t \right) + \right. \\
 & + \frac{2q''_\chi(u) \operatorname{tr} \left[C_t \left(x - \hat{X}_t \right) \left(x^T - \hat{X}_t^T \right) C_t h \sigma_2 h^T \right]}{\mu_t^2} + \\
 & + \frac{q'_\chi(u) \operatorname{tr} \left[C_t h \sigma_2 h^T \right]}{\mu_t^2} - \frac{4q''_\chi(u) \hat{X}_t^T C_t \left(x - \hat{X}_t \right) \left(x^T - \hat{X}_t^T \right) C_t h \sigma_2 b^T}{\mu_t^2} - \\
 & - \frac{2q'_\chi(u) \hat{X}_t^T C_t h \sigma_2 b^T}{\mu_t^2} + \frac{2q'_\chi(u) \left(x^T - \hat{X}_t^T \right) C_t h \sigma_2 b^T}{\mu_t^2} + \\
 & + \frac{2q''_\chi(u) \hat{X}_t^T C_t \left(x - \hat{X}_t \right) \left(x^T - \hat{X}_t^T \right) C_t \hat{X}_t b \sigma_2 b^T}{\mu_t^2} - \\
 & - \frac{2q'_\chi(u) \hat{X}_t^T C_t \left(x - \hat{X}_t \right) b \sigma_2 b^T}{\mu_t^2} + \frac{q'_\chi(u) \hat{X}_t^T C_t \hat{X}_t b \sigma_2 b^T}{\mu_t^2} \Big\} dt - \\
 & - \frac{2q'_\chi(u) (x^T - \hat{X}_t^T) C_t dm_t}{\mu_t} + \frac{2q'_\chi(u) \hat{X}_t^T C_t \left(x - \hat{X}_t \right) d\mu_t}{\mu_t}. \quad (12)
 \end{aligned}$$

Далее найдем дифференциал Ито произведения $q_\chi(u)\tilde{g}_t(\lambda)$. По формуле для дифференциала Ито произведения двух процессов получим:

$$\begin{aligned}
 d[q_\chi(u)\tilde{g}_t(\lambda)] = & \\
 = q_\chi(u) d\tilde{g}_t(\lambda) + dq_\chi(u)\tilde{g}_t(\lambda) + & \frac{2q'_\chi(u)(x^T - \hat{X}_t^T)C_t(h + \hat{X}_t b)\sigma_2 B^T dt}{\mu_t}. \quad (13)
 \end{aligned}$$

Вычислим каждое из трех слагаемых в (13) отдельно, заменив предварительно переменные x и u операторами $\partial/i\partial\lambda$ и U_λ соответственно.

Сначала находим первое слагаемое:

$$\begin{aligned}
 [q_\chi(u) d\tilde{g}_t(\lambda)]_{\lambda=0} = & M_{\Delta^U}^{\tilde{p}_t^*} \left\{ q'_\chi(u) 2\varphi^T C_t \left(x - \hat{X}_t \right) + \right. \\
 & + \operatorname{tr} (C_t \sigma_0) + 2q''_\chi(u) \left(x^T - \hat{X}_t^T \right) C_t \sigma_0 C_t \left(x - \hat{X}_t \right) +
 \end{aligned}$$

$$\begin{aligned}
 & + \int_{R_0^q} \left[q_\chi(\bar{u}) - q_\chi(u) - 2q'_\chi(u) \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t \psi'' \right] \nu_P(t, dv) \Bigg\} dt + \\
 & + M_{\Delta_U}^{\tilde{p}_t^*} \left\{ q_\chi(u) \varphi_1^\text{T} + q'_\chi(u) \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t \sigma_1 \right\} \frac{dY_t}{\sigma_2}, \quad (14)
 \end{aligned}$$

где $\sigma_0 = b\nu_0 b^\text{T}$;

$$\bar{u} = \left(x^\text{T} + \psi''^\text{T} - \hat{X}_t^\text{T} \right) C_t \left(x + \psi'' - \hat{X}_t \right).$$

Для вычисления второго слагаемого в правой части (13) заметим, что для любого полинома $p(X)$ в силу (14) имеем:

$$\left[p \left(\frac{\partial}{i\partial\lambda} \right) \tilde{g}_t(\lambda) \right]_{\lambda=0} = M_{\Delta_x}^p [\tilde{p}_t].$$

Пользуясь аппроксимацией (8) ненормированной плотности $\tilde{p}_t(x)$, получаем:

$$\begin{aligned}
 [dq_\chi(U_\lambda)\tilde{g}_t(\lambda)]_{\lambda=0} = M_{\Delta_x}^{p^*} \Bigg\{ & \left[q'_\chi(u) \left(x^\text{T} - \hat{X}_t^\text{T} \right) \dot{C}_t(x - \hat{X}_t) + \right. \\
 & + \frac{2q''_\chi(u) \text{tr} \left[C_t \left(x - \hat{X}_t \right) \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t h \sigma_2 h^\text{T} \right]}{\mu_t^2} + \\
 & + \frac{q'_\chi \text{tr} \left(C_t h \sigma_2 h^\text{T} \right)}{\mu_t^2} - \frac{4q''_\chi(u) \hat{X}_t^\text{T} C_t \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t h \sigma_2 b^\text{T}}{\mu_t^2} - \\
 & - \frac{2q'_\chi(u) \hat{X}_t^\text{T} C_t h \sigma_2 b^\text{T}}{\mu_t^2} + \frac{2q'_\chi(u) (x^\text{T} - \hat{X}_t^\text{T}) C_t h \sigma_2 b^\text{T}}{\mu_t^2} + \\
 & + \frac{2q''_\chi(u) \hat{X}_t^\text{T} C_t \left(x - \hat{X}_t \right) \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t \hat{X}_t b \sigma_2 b^\text{T}}{\mu_t^2} - \\
 & - \frac{2q'_\chi(u) \hat{X}_t^\text{T} C_t \left(x - \hat{X}_t \right) b \sigma_2 b^\text{T}}{\mu_t^2} + \frac{q'_\chi(u) \hat{X}_t^\text{T} C_t \hat{X}_t b \sigma_2 b^\text{T}}{\mu_t^2} \Bigg] dt - \\
 & - \frac{2q'_\chi(u) \left(x^\text{T} - \hat{X}_t^\text{T} \right) C_t dm_t}{\mu_t} + \frac{2q'_\chi(u) \hat{X}_t^\text{T} C_t \left(x - \hat{X}_t \right) d\mu_t}{\mu_t} \Bigg\}. \quad (15)
 \end{aligned}$$

Имея в виду, что μ и матрицы C_t , h , b , $\sigma_2 = \psi_1 \nu \psi_1^T$ и \hat{X}_t от X не зависят и $(X^T - \hat{X}_t^T) \dot{C}_t (X - \hat{X}_t) = \text{tr} [\dot{C}_t (X - \hat{X}_t) (x^T - \hat{X}_t^T)]$, приходим к выводу, что при аппроксимации (8) функции $\tilde{p}_t(x)$ необходимо вычислить четыре типа интегралов (напомним, что $p_0(u) = 1$):

$$\begin{aligned} I_1 &= M_{\Delta^x}^{wp_\nu} \left[q'_\chi(U) (X - \hat{X}_t) \right]; \\ I_2 &= M_{\Delta^x}^{wp_\nu} \left[q'_\chi(U) (X - \hat{X}_t) (X^T - \hat{X}_t^T) \right]; \\ I_3 &= M_{\Delta^x}^{wp_\nu} \left[q''_\chi(U) (X - \hat{X}_t) (X^T - \hat{X}_t^T) \right]; \\ I_4 &= M_{\Delta^x}^{wp_\nu} \left[q'_\chi(U) (X - \hat{X}_t) \right]. \end{aligned}$$

3.5. Следуя [6], нетрудно проверкой убедиться, что $I_1 = 0$, а

$$\begin{aligned} I_2 &= \int_{-\infty}^{\infty} q'_\chi(u) (x - \hat{X}_t) (x^T - \hat{X}_t^T) w(u) p_\nu(u) dx = \\ &= \frac{K_t}{2^{n/2+1} \Gamma(n/2 + 1)} \int_{-\infty}^{\infty} q'_\chi(u) p_\nu(u) u^{n/2} e^{-u/2} du \quad (16) \end{aligned}$$

(для гауссовой плотности $w(u)$). Так же вычисляется интеграл

$$\begin{aligned} I_3 &= \int_{-\infty}^{\infty} q''_\chi(u) (x - \hat{X}_t) (x^T - \hat{X}_t^T) w(u) p_\nu(u) dx = \\ &= \frac{K_t}{2^{n/2+1} \Gamma(n/2 + 1)} \int_{-\infty}^{\infty} q''_\chi(u) p_\nu(u) u^{n/2} e^{-u/2} du. \quad (17) \end{aligned}$$

Вследствие ортогональности степеней u^λ полиному $p_\nu(u)$ с весом

$$w(u) = \frac{1}{2^{n/2} \Gamma(n/2)} u^{n/2-1} e^{-u/2}$$

при $\lambda < \nu$ интеграл в (16) равен нулю при всех $\nu > \chi$. Нормальному распределению $\mathcal{N}(0, K_t)$, $K_t = C_t^{-1}$, n -мерного случайного вектора X_t соответствует χ^2 -распределение с n степенями свободы квадратичной формы $U_t = X_t^T C_t X_t$.

По той же причине и в силу биортонормальности полиномов $p_\nu(u)$ и $q_\nu(u)$ при $\nu = \chi$ находим из (17), что

$$I_3 = \frac{K_t \chi}{n}.$$

Аналогично путем интегрирования по частям убеждаемся в том, что интеграл в (16) равен нулю при всех $\nu < \chi - 1$, а при $\nu = \chi - 1$ для I_2 имеем

$$I_2 = \frac{K_t}{2n}.$$

Интеграл в (16) равен нулю при всех $\nu > \chi - 1$, а при $\nu = \chi - 1$

$$I_2 = \frac{K_t(\chi - 1)}{2n(n + 2\chi - 2)}.$$

При $\nu < \chi - 1$ интегрированием по частям (17) находим:

$$\begin{aligned} \int_0^\infty q_\chi''(u)p_\nu(u)u^{n/2}e^{-u/2}du &= - \int_0^\infty q_\chi'(u)p_\nu'(u)u^{n/2}e^{-u/2}du - \\ &- \frac{n}{2} \int_0^\infty q_\chi'(u)p_\nu(u)u^{n/2-1}e^{-u/2}du + \frac{1}{2} \int_0^\infty q_\chi'(u)p_\nu(u)u^{n/2}e^{-u/2}du = \\ &= \int_0^\infty q_\chi(u)p_\nu''(u)u^{n/2}e^{-u/2}du + \frac{n}{2} \int_0^\infty q_\chi(u)p_\nu'(u)u^{n/2-1}e^{-u/2}du - \\ &- \frac{1}{2} \int_0^\infty q_\chi(u)p_\nu'(u)u^{n/2}e^{-u/2}du - \frac{n}{2} \int_0^\infty q_\chi'(u)p_\nu(u)u^{n/2-1}e^{-u/2}du + \\ &+ \frac{1}{2} \int_0^\infty q_\chi'(u)p_\nu(u)u^{n/2}e^{-u/2}du. \end{aligned}$$

Первые три интеграла в последней части этих равенств равны нулю вследствие ортогональности $q_\chi(u)$ степеням u^λ при $\lambda \leq \nu - 1 < \chi - 2$. Последний интеграл тоже равен нулю при $\nu < \chi - 1$.

Следовательно, при $\nu < \chi - 1$

$$I_3 = -\frac{K_t}{2^{n/2+1}\Gamma(n/2)} \int_0^\infty q_\chi'(u)p_\nu(u)u^{n/2-1}e^{-u/2}du = -\frac{1}{2} K_t I_4.$$

Отсюда при подстановке в (15) выражения функции $\tilde{p}_t^*(u)$ (8) все выражения, содержащие интегралы I_3 и I_4 , сократятся, кроме соответствующих значению $\nu = \chi - 1$. Интеграл I_4 при $\nu = \chi - 1$ имеет вид:

$$I_4 = -\frac{1}{2^{n/2}\Gamma(n/2)} \int_0^\infty q'_\chi(u)p_{\chi-1}(u)u^{n/2-1}e^{-u/2}du = -\frac{1}{2(n+2\chi-2)}.$$

В результате формулу (15) можно представить в следующем виде:

$$\begin{aligned} [dq_\chi(U_\lambda)\tilde{g}_t(\lambda)]_{\lambda=0} &= \left\{ \frac{1}{n} \left(\frac{c_{\chi-1}}{2} + \chi c_\chi \right) \text{tr} (\dot{C}_t K_t) + \right. \\ &\quad \left. + \frac{c_{\chi-1}}{2n} \cdot \frac{\text{tr} [C_t h \sigma_2 h^T] - 2\hat{X}_t^T C_t h \sigma_2 b^T + \hat{X}_t^T C_t b \sigma_2 b^T}{\mu_t^2} \right\} dt. \end{aligned} \quad (18)$$

Вычислим последнее слагаемое в (13) при $x = \partial/i\partial\lambda$, $u = U_\lambda$ и $\lambda = 0$. Учтя, что от λ в этом слагаемом зависит только

$$B = M_{\Delta^x}^{p^*} \left\{ \left[(\varphi_1^T + i\lambda^T \sigma_1) e^{i\lambda^T X} \right] \sigma_2^{-1} \right\},$$

и выполнив дифференцирование, совершенно так же, как при выводе формулы (14), получим при аппроксимации (8) функции $\tilde{p}_t(x)$ следующее выражение:

$$\begin{aligned} &\left[q'_\chi(U_\lambda) \left(\frac{\partial^T}{i\partial\lambda} - \hat{X}_t^T \right) C_t (h + \hat{X}_t b) \sigma_2 B^T \right]_{\lambda=0} = \\ &= M_{\Delta^x}^{p^*} \left\{ q'_\chi(U_\lambda) \left(X^T - \hat{X}_t^T \right) C_t (h + \hat{X}_t b) \varphi_1 + \right. \\ &+ 2q''_\chi(U_\lambda) \text{tr} \left[(h + \hat{X}_t b) \sigma_1^T C_t (X - \hat{X}_t) (X^T - \hat{X}_t^T) C_t \right] + \\ &\quad \left. + q'_\chi(U_\lambda) \text{tr} \left[(h + \hat{X}_t b) \sigma_1^T C_t \right] \right\} \frac{dt}{\mu}. \end{aligned} \quad (19)$$

3.6. Подставив выражения (14), (18) и (19) в (12) при $x = \partial/i\partial\lambda$, $u = U_\lambda$, после замены индекса ν на χ получаем уравнение:

$$\begin{aligned} dc_\chi &= M_{\Delta^x}^{p^*} \left\{ q'_\chi(u) (2\varphi^T C_t (X - \hat{X}_t) + \text{tr} [C_t \sigma_0]) + \right. \\ &\quad \left. + 2q''_\chi(u) (X^T - \hat{X}_t^T) C_t \sigma_0 C_t (X - \hat{X}_t) + \right. \\ &\quad \left. + q'_\chi(U_\lambda) \text{tr} \left[(h + \hat{X}_t b) \sigma_1^T C_t \right] \right\} \frac{dt}{\mu}. \end{aligned}$$

$$\begin{aligned}
 & + \int_{R_0^q} \left[q_\chi(\bar{u}) - q_\chi(u) - 2q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \psi'' \right] \nu_P(t, dv) - \\
 & - \frac{q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \left(h + \hat{X}_t b \right) \varphi_1}{\mu_t} + \frac{q'_\chi(u) \operatorname{tr} \left[\left(h + \hat{X}_t b \right) \sigma_1^T C_t \right]}{\mu_t} + \\
 & + \frac{2q''_\chi(u) \operatorname{tr} \left[\left(h + \hat{X}_t b \right) \sigma_1^T C_t \left(X - \hat{X}_t \right) \left(X^T - \hat{X}_t^T \right) C_t \right]}{\mu_t} \Bigg\} dt + \\
 & + \left\{ \frac{1}{2n} (c_{\chi-1} + 2\chi c_\chi) \operatorname{tr} \left[\dot{C}_t K_t \right] + \right. \\
 & \left. + \frac{c_{\chi-1}}{2n} \cdot \frac{\operatorname{tr} \left[C_t h \sigma_2 h^T \right] - 2\hat{X}_t^T C_t h \sigma_2 b^T + \hat{X}_t^T C_t \hat{X}_t b \sigma_2 b^T}{\mu_t^2} \right\} dt + \\
 & + M_{\Delta_x}^{p^*} \left\{ \frac{q_\chi(u) \varphi_1^T + q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \sigma_1}{\sigma_2} \right\} dY_t. \quad (20)
 \end{aligned}$$

Примем в дополнение к обозначениям (11) следующие:

$$\begin{aligned}
 \gamma_{\chi 0} &= \gamma_{\chi 0} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= M_{\Delta_x}^w \left\{ q'_\chi(u) \left(2\varphi(X, Y_t, \Theta, t)^T C_t \left(X - \hat{X}_t \right) + \operatorname{tr} [C_t \sigma_0(X, Y_t, \Theta, t)] \right) + \right. \\
 &\quad \left. + 2q''_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \sigma_0(X, Y_t, \Theta, t) C_t \left(X - \hat{X}_t \right) + \right. \\
 &\quad \left. + \int_{R_0^q} \left[q_\chi(\bar{u}) - q_\chi(u) - 2q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \psi''(X, Y_t, \Theta, t, v) \right] \nu_P(t, dv) \right\}; \\
 \gamma_{\chi \nu} &= \gamma_{\chi \nu} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= M_{\Delta_x}^{wp\nu} \left\{ q'_\chi(u) \left(2\varphi(X, Y_t, \Theta, t)^T C_t \left(X - \hat{X}_t \right) + \operatorname{tr} [C_t \sigma_0(X, Y_t, \Theta, t)] \right) + \right. \\
 &\quad \left. + 2q''_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \sigma_0(X, Y_t, \Theta, t) C_t \left(X - \hat{X}_t \right) + \right. \\
 &\quad \left. + \int_{R_0^q} \left[q_\chi(\bar{u}) - q_\chi(u) - 2q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \psi''(X, Y_t, \Theta, t, v) \right] \nu_P(t, dv) \right\};
 \end{aligned}$$

$$\begin{aligned}
 \varepsilon_{\chi 0} &= \varepsilon_{\chi 0} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= M_{\Delta^x}^w \left\{ q'_\chi(u) \left[\sigma_1 \left(X, Y_t, \Theta, t \right)^T - \varphi_1(X, Y_t, \Theta, t) \left(X^T - \hat{X}_t^T \right) \right] + \right. \\
 &\quad \left. + 2q''_\chi(u) \sigma_1(X, Y_t, \Theta, t)^T C_t \left(X - \hat{X}_t \right) \left(X^T - \hat{X}_t^T \right) \right\}; \\
 \varepsilon_{\chi \nu} &= \varepsilon_{\chi \nu} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= M_{\Delta^x}^{wp_\nu} \left\{ q'_\chi(u) \left[\sigma_1 \left(X, Y_t, \Theta, t \right)^T - \varphi_1(X, Y_t, \Theta, t) \left(X^T - \hat{X}_t^T \right) \right] + \right. \\
 &\quad \left. + 2q''_\chi(u) \sigma_1(X, Y_t, \Theta, t)^T C_t \left(X - \hat{X}_t \right) \left(X^T - \hat{X}_t^T \right) \right\}; \\
 \eta_{\chi 0} &= \eta_{\chi 0} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= \frac{M_{\Delta^x}^w \left\{ q_\chi(u) \varphi_1(X, Y_t, \Theta, t)^T + q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \sigma_1(X, Y_t, \Theta, t) \right\}}{\sigma_2(Y_t, \Theta, t)}; \\
 \eta_{\chi \nu} &= \eta_{\chi \nu} \left(Y_t, \hat{X}_t, \Theta, t \right) = \\
 &= \frac{M_{\Delta^x}^{wp_\nu} \left\{ q_\chi(u) \varphi_1(X, Y_t, \Theta, t)^T + q'_\chi(u) \left(X^T - \hat{X}_t^T \right) C_t \sigma_1(X, Y_t, \Theta, t) \right\}}{\sigma_2(Y_t, \Theta, t)}.
 \end{aligned}$$

Тогда, имея в виду, что $u^T F v = \text{tr}[F v u^T]$ для любой квадратной матрицы F и векторов u и v , можем переписать уравнения (20) в виде:

$$\begin{aligned}
 dc_\chi &= \left\{ \mu_t \gamma_{\chi 0} \left(Y_t, \hat{X}_t, \Theta, t \right) + \sum_{\nu=1}^N c_\nu \gamma_{\chi \nu} \left(Y_t, \hat{X}_t, \Theta, t \right) + \right. \\
 &\quad + \text{tr} \left[\left\{ \mu_t \left(h_0 \left(Y_t, \hat{X}_t, \Theta, t \right) + \hat{X}_t b_0 \left(Y_t, \hat{X}_t, \Theta, t \right) \right) \right\} + \right. \\
 &\quad + \sum_{\nu=1}^N c_\nu \left(h_\nu \left(Y_t, \hat{X}_t, \Theta, t \right) + \hat{X}_t b_\nu \left(Y_t, \hat{X}_t, \Theta, t \right) \right) \times \\
 &\quad \times \left. \left\{ \varepsilon_{\chi 0} \left(Y_t, \hat{X}_t, \Theta, t \right) + \sum_{\nu=1}^N \frac{c_\nu \varepsilon_{\chi \nu} \left(Y_t, \hat{X}_t, \Theta, t \right)}{\mu_t} \right\} C_t \right] + \\
 &\quad \left. + \frac{1}{2n} (c_{\chi-1} + 2\chi c_\chi) \text{tr} [\dot{C}_t K_t] \right\}
 \end{aligned}$$

$$\begin{aligned}
 & + \frac{c_{\chi-1}}{2n} \operatorname{tr} \left[C_t \left(h_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N \frac{c_\nu h_\nu(Y_t, \hat{X}_t, t)}{\mu_t} \right) \times \right. \\
 & \quad \times \sigma_2(Y_t, \Theta, t) \left(h_0(Y_t, \hat{X}_t, \Theta, t)^T + \sum_{\nu=1}^N \frac{c_\nu h_\nu(Y_t, \hat{X}_t, \Theta, t)^T}{\mu_t} \right) \Big] - \\
 & \quad - 2\hat{X}_t^T C_t \left(h_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N \frac{c_\nu h_\nu(Y_t, \hat{X}_t, \Theta, t)}{\mu_t} \right) \times \\
 & \quad \times \sigma_2(Y_t, \Theta, t) \left(b_0(Y_t, \hat{X}_t, \Theta, t)^T + \sum_{\nu=1}^N \frac{c_\nu b_\nu(Y_t, \hat{X}_t, \Theta, t)^T}{\mu_t} \right) + \\
 & \quad + \hat{X}_t^T C_t \hat{X}_t \left(b_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N \frac{c_\nu b_\nu(Y_t, \hat{X}_t, \Theta, t)}{\mu_t} \right) \times \\
 & \quad \times \sigma_2(Y_t, \Theta, t) \left(b_0(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N \frac{c_\nu b_\nu(Y_t, \hat{X}_t, \Theta, t)^T}{\mu_t} \right) \Big\} dt + \\
 & \quad + \left\{ \mu_t \eta_{\chi 0}(Y_t, \hat{X}_t, \Theta, t) + \sum_{\nu=1}^N c_\nu \eta_{\chi \nu}(Y_t, \hat{X}_t, \Theta, t) \right\} dY_t \\
 & \quad (\chi = 1, \dots, N). \quad (21)
 \end{aligned}$$

3.7. Уравнения (9), (10), (21) и соотношение $\hat{X}_t = m_t/\mu_t$ при начальных условиях

$$m(t_0) = \mathbb{M}[X_0 | Y_0]; \quad \mu(t_0) = 1; \quad c_\chi(t_0) = c_{\chi 0} \quad (\chi = 1, \dots, N) \quad (22)$$

$(c_{\chi 0} \quad (\chi = 1, \dots, N))$ — коэффициенты разложения (8) условной плотности вероятности $\tilde{p}_{t_0}(x) = p_0(x | Y_0)$ вектора X_0 относительно Y_0 определяют МЭСОФ. После решения уравнений (10) и (21) с.к. оптимальная оценка вектора состояния и ковариационная матрица ошибки фильтрации в МЭСОФ определяются по следующим приближенным формулам:

$$\hat{X}_t = \frac{m_t}{\mu_t}; \quad (23)$$

$$R_t = M_{\Delta_x}^w \left[\left(X - \frac{m_t}{\mu_t} \right) \left(X^T - \frac{m_t^T}{\mu_t} \right) \right] + \\ + \sum_{\nu=1}^N \frac{c_\nu}{\mu_t} M_{\Delta_x}^{wp_\nu} \left[\left(X - \frac{m_t}{\mu_t} \right) \left(X^T - \frac{m_t^T}{\mu_t} \right) \right]. \quad (24)$$

Порядок МЭСОФ особенно при большой размерности n вектора состояния системы значительно ниже порядка других СОФ. Так, при учете в разложении (10) моментов до десятого порядка, когда уже при $n > 3$, $N = 5$ имеем $n + N + 1 \leq n(n + 3)/2$. При $n > 3$, $N = 5$ ЭСОФ имеет более низкий порядок, чем фильтры МНА, обобщенный фильтр Калмана–Бьюси, фильтры второго порядка и гауссовский фильтр.

Таким образом, в основе алгоритма эллипсоидальной субоптимальной нелинейной фильтрации лежат следующие утверждения.

Теорема 3.1. *В условиях теоремы 2.1, если ЭСОФ существует, то он определяется уравнениями (9), (10) и (21) при условиях (22)–(24).*

Теорема 3.2. *В условиях теоремы 2.2, если ЭСОФ существует, то он определяется уравнениями теоремы 3.1 при условиях (7).*

3.8. Применяя методы теории чувствительности для приближенного анализа фильтрационных уравнений и учитывая случайность параметров Θ , приDEM к следующим уравнениям для функций чувствительности первого порядка [2]:

$$\begin{aligned} d\nabla^\Theta \hat{X}_s &= \nabla^\Theta A^{\hat{X}_s} dt + \nabla^\Theta B^{\hat{X}_s} dY_t, \quad \nabla^\Theta B^{\hat{X}_s}(t_0) = 0; \\ d\nabla^\Theta R_{sq} &= \nabla^\Theta A^{R_{sq}} dt + \nabla^\Theta B^{R_{sq}} dY_t, \quad \nabla^\Theta R_{sq}(t_0) = 0; \\ d\nabla^\Theta c_\kappa &= \nabla^\Theta A^{c_\kappa} dt + \nabla^\Theta B^{c_\kappa} dY_t, \quad \nabla^\Theta c_\kappa(t_0) = 0. \end{aligned}$$

Здесь процедура взятия производных осуществляется по всем входящим переменным, а коэффициенты чувствительности вычисляются при $\Theta = m^\Theta$. При этом предполагается малость дисперсий по сравнению с их математическими ожиданиями. Очевидно, что при дифференцировании по Θ ($\nabla^\Theta = \partial/\partial\Theta$) порядок уравнений возрастает пропорционально числу производных. Аналогично составляются уравнения для элементов матриц вторых функций чувствительности.

Для оценки качества ЭСОФ, следя [1, 2], при гауссовых Θ с математическим ожиданием m^Θ и ковариационной матрицей K^Θ введем условную функцию потерь, допускающую квадратичную аппроксимацию:

$$\rho^{\hat{X}_s} = \rho^{\hat{X}_s}(\Theta) = \rho(m^\Theta) + \sum_{ii=1}^{n^\Theta} \rho'_i(m^\Theta) \Theta_s^0 + \sum_{i,j=1}^{n^\Theta} \sum_{j=1}^{n^\Theta} \rho''_{ij}(m^\Theta) \Theta_i^0 \Theta_j^0,$$

а также показатель ε

$$\varepsilon = \varepsilon_2^{1/4}.$$

Здесь введены обозначения:

$$\varepsilon_2 = M^{\Theta A} [\rho(\Theta)^2] - \rho(m^\Theta)^2;$$

$$M^{\Theta A} [\rho(\Theta)^2] = \rho(m^\Theta)^2 + \rho'(m^\Theta)^T K^\Theta \rho'(m^\Theta) +$$

$$+ 2\rho(m^\Theta) \operatorname{tr} [\rho''(m^\Theta) K^\Theta] + \left\{ \operatorname{tr} [\rho''(m^\Theta) K^\Theta] \right\}^2 + 2 \operatorname{tr} [\rho''(m^\Theta) K^\Theta]^2,$$

а функции ρ' и ρ'' по известным формулам [1, 2] определяются на основе первых и вторых функций чувствительности.

Изложенные выше методы синтеза ЭСОФ дают принципиальную возможность получить фильтр, близкий к оптимальному по оценке с любой степенью точности. Чем выше максимальный порядок учитываемых моментов, коэффициент эллипсоидальной аппроксимации, тем выше будет точность приближения к оптимальной оценке. Однако число уравнений, определяющих параметры апостериорного одномерного эллипсоидального распределения, быстро растет с увеличением числа учитываемых параметров.

4 Заключение

Разработана теория аналитического синтеза МЭСОФ для нелинейных дифференциальных МСтС (3)–(4) и (6). Рассмотрены случаи гауссовских и негауссовских СтС. Алгоритмы ЭСОФ положены в основу модуля экспериментального программного обеспечения «StS-Filter» (version 2016).

Результаты допускают обобщение на случай дискретных СтС, а также авторелизованных МСтС.

Теоретический и практический интерес представляет теория ЭСОФ для МСтС (1) и (2).

Литература

1. Синицын И. Н. Ортогональные субоптимальные фильтры для нелинейных стохастических систем на многообразиях // Информатика и её применения, 2016. Т. 10. Вып. 1. С. 34–44.
2. Синицын И. Н., Синицын В. И., Корепанов Э. Р. Эллипсоидальные субоптимальные фильтры для нелинейных стохастических систем на многообразиях // Информатика и её применения, 2016 (в печати). Т. 10. Вып. 2.
3. Синицын И. Н., Синицын В. И. Лекции по теории нормальной и эллипсоидальной аппроксимации распределений в стохастических системах. — М.: ТОРУС ПРЕСС, 2013. 488 с.
4. Пугачёв В. С., Синицын И. Н. Теория стохастических систем. — М.: Логос, 2000; 2004. 1000 с.

5. Справочник по теории вероятностей и математической статистике / Под ред. В. С. Королюка, Н. И. Портенко, А. В. Скорохода, А. Ф. Турбина. — М.: Наука, 1985. 640 с.
6. Синицын И. Н. Фильтры Калмана и Пугачёва. — 2-е изд. — М.: Логос, 2007. 776 с.
7. Wonham M. Some application of stochastic differential equations to optimal nonlinear filtering // J. Soc. Ind. Appl. Math. Control, 1965. Vol. 2. P. 347–369.
8. Zakai M. On the optimal filtering of diffusion processes // Ztschr. Wahrscheinlichkeitstheorie Verw. Geb., 1969. Bd. 11. S. 230–243.

Поступила в редакцию 04.03.16

MODIFIED ELLIPSOIDAL SUBOPTIMAL FILTERS FOR NONLINEAR STOCHASTIC SYSTEMS ON MANIFOLDS

I. N. Sinitsyn, V. I. Sinitsyn, and E. R. Korepanov

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: The analytical synthesis theory for modified ellipsoidal suboptimal filters (MESOF) for nonlinear stochastic systems on manifolds (MStS) based on the nonnormed *a posteriori* characteristic function is developed. Gaussian and non-Gaussian MStS are considered. Symbolic MESOF algorithms are more simple than the ESOF algorithms. The MESOF algorithms are the basis of the software tools (“StS-Filter,” 2016).

Keywords: accuracy and sensitivity equations; ellipsoidal approximation method (EAM); ellipsoidal suboptimal filter (ESOF); modified ellipsoidal suboptimal filter (MESOF); nonnormed characteristic function; Poisson noise; suboptimal filter (SOF); Wiener noise

DOI: 10.14357/08696527160205

Acknowledgments

The research was supported by the Russian Foundation for Basic Research (project 15-07-002244).

References

1. Sinitsyn, I. N. 2016. Ortogonal'nye suboptimal'nye fil'try dlya nelineynykh stokhasticheskikh sistem na mnogoobraziyakh [Orthogonal suboptimal filters for nonlinear stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 10(1):34–44.

2. Sinitsyn, I. N., V. I. Sinitsyn, and E. R. Korepanov. 2016 (in press). Ellipsoidal'nye suboptimal'nye fil'try dlya nelineynykh stokhasticheskikh sistem na mnogoobraziyakh [Ellipsoidal suboptimal filters for nonlinear stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 10(2).
3. Sinitsyn, I. N., and V. I. Sinitsyn. 2013. Lektsii po teorii normal'noy i ellipsoidal'noy approksimatsii raspredeleniy v stokhasticheskikh sistemakh [Lectures on normal and ellipsoidal approximations of distributions in stochastic systems]. Moscow: TORUS PRESS. 488 p.
4. Pugachev, V. S., and I. N. Sinitsyn. 2001. *Stochastic systems. Theory and applications*. Singapore: World Scientific. 908 p.
5. Korolyuk, V. S., N. I. Portenko, A. V. Skorokhod, and A. F. Turbin, eds. 1985. *Spravochnik po teorii veroyatnosti i matematicheskoy statistike* [Handbook: Probability theory and mathematical statistics]. Moscow: Nauka. 640 p.
6. Sinitsyn, I. N. 2007. *Fil'try Kalmana i Pugacheva* [Kalman and Pugachev filters]. 2nd ed. Moscow: Logos. 776 p.
7. Wonham, M. 1965. Some application of stochastic differential equations to optimal nonlinear filtering. *J. Soc. Ind. Appl. Math. Control* 2:347–369.
8. Zakai, M. 1969. On the optimal filtering of diffusion processes. *Ztschr. Wahrscheinlichkeitstheorie Verw. Geb.* 11:230–243.

Received March 4, 2016

Contributors

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sinitsin@dol.ru

Sinitsyn Vladimir I. (b. 1968) — Doctor of Science in physics and mathematics, associate professor, Head of Department, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; VSinitsyn@ipiran.ru

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Ekorepanov@ipiran.ru

СОЗДАНИЕ РЕАЛИСТИЧНЫХ НАБОРОВ ДАННЫХ ДЛЯ АЛГОРИТМОВ ТРЕХМЕРНОЙ РЕКОНСТРУКЦИИ С ПОМОЩЬЮ ВИРТУАЛЬНОЙ СЪЕМКИ КОМПЬЮТЕРНОЙ МОДЕЛИ

O. A. Яковлев¹, A. B. Гасилов²

Аннотация: Рассмотрен способ создания наборов данных для алгоритмов разреженной реконструкции путем виртуальной съемки сцены, представленной компьютерной моделью. Виртуальная съемка позволяет в интерактивном режиме делать снимки с различных ракурсов в соответствии с моделью проекционной камеры. Результатом виртуальной съемки является набор снимков, каждый из которых описывается параметрами камеры и списком наблюдаемых точек. Такой набор данных может подаваться на вход программам, реализующим алгоритмы реконструкции, значительно упрощая их тестирование и оценку качества. Реалистичность результата обеспечивается в соответствии с наиболее значимыми характеристиками реальных наборов данных. Отмечены особенности реализации виртуальной съемки с использованием библиотеки OpenCV и представлено программное обеспечение (ПО), разработанное в соответствии с изложенным материалом.

Ключевые слова: трехмерная реконструкция; искусственные данные; виртуальная съемка; OpenCV

DOI: 10.14357/08696527160206

1 Введение

Задача трехмерной (3D) реконструкции состоит в установлении трехмерной структуры сцены по набору снимков, сделанных с различных ракурсов. В данной работе рассматривается задача разреженной реконструкции (structure from motion, SfM) [1], результатом решения которой является облако точек, принадлежащих сцене. Все алгоритмы реконструкции включают предварительный этап — выделение особых точек на снимках, поиск соответствий между этими точками и построение графа сцены. Под графом сцены понимается неориентированный граф, вершинами которого служат особые точки на снимках, а ребра показывают соответствия между этими точками. Так как этап построения графа сцены явля-

¹Орловский филиал Федерального исследовательского центра «Информатика и управление» Российской академии наук, tmausgra@gmail.com

²Орловский филиал Федерального исследовательского центра «Информатика и управление» Российской академии наук, gasilov.av@ya.ru

ется общим для всех алгоритмов, можно утверждать, что граф сцены и служит исходным материалом для алгоритмов реконструкции.

При разработке, модификации или исследовании различных алгоритмов реконструкции возникают такие задачи, как тестирование и оценка качества реконструкции.

Существует множество наборов данных для тестирования алгоритмов реконструкции, представленных коллекциями снимков. Однако процесс выделения особых точек на снимках и поиска соответствий между ними остается чрезвычайно затратным с точки зрения вычислительных ресурсов и представляет собой наиболее трудоемкий этап в алгоритмах, реализующих последовательный принцип реконструкции [2].

Для оценки качества прежде всего необходим эталон реконструкции. В задаче многовидового стереозрения [3], результатом решения которой является полигональная или воксельная модель поверхности, получить эталон можно с помощью специального оборудования. В работе [4] описана методика оценки качества реконструкции в задаче многовидового стереозрения, а также авторы предоставили два набора данных и эталоны реконструкции для них, полученные с помощью 3D-сканера. Построение эталона в задаче разреженной реконструкции даже при наличии 3D-сканера значительно усложняется тем, что состав облака точек главным образом определяется алгоритмом выделения и сопоставления особых точек.

Изложенные аргументы приводят к выводу о том, что проблема тестирования и оценки качества реконструкции может быть решена путем использования искусственных наборов данных, созданных с помощью виртуальной съемки компьютерных моделей. В данной работе детально рассматривается задача генерации реалистичных наборов данных путем съемки полигональных моделей, приводятся алгоритмы решения сопутствующих задач, а также особенности практической реализации с использованием библиотеки OpenCV.

2 Модель камеры

Процесс формирования снимка наилучшим образом описывается моделью камеры-обскуры, или проективной камеры. Согласно модели камеры-обскуры образ точки (X, Y, Z) , имеющей координаты (x, y) на снимке, получается путем центрального проецирования с центром проекции, совпадающим с оптическим центром камеры, т. е.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim K[R t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (1)$$

где K — матрица внутренних параметров камеры; R и t — матрица поворота и вектор переноса соответственно, описывающие преобразование из мировой си-

стемы координат (СК) в СК камеры; $[R t]$ — слияние матрицы и вектора путем добавления столбца. Символ \sim обозначает отношение эквивалентности в проективном пространстве. Матрица $[R t]$ описывает положение камеры относительно сцены и называется матрицей внешних параметров.

Линзы камеры вносят нелинейные искажения в получаемый снимок — радиальные дисторсии. Для моделирования современных камер ограничиваются двумя порядками дисторсий. Дисторсия является нелинейным преобразованием, для ее записи необходимо раскрыть выражение (1). Пусть

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t,$$

тогда

$$x = \frac{f_x u}{w} (1 + k_1 r^2 + k_2 r^4) + c_x; \quad y = \frac{f_y v}{w} (1 + k_1 r^2 + k_2 r^4) + c_y,$$

где $r = (1/w)\sqrt{u^2 + v^2}$; k_1 и k_2 — коэффициенты дисторсии; f_x , f_y , c_x и c_y — внутренние параметры камеры.

Модуль Viz библиотеки OpenCV позволяет визуализировать центральную проекцию трехмерной сцены в соответствии с моделью проективной камеры. Для описания текущего вида сцены модуль Viz использует концепцию наблюдателя. Наблюдатель, подобно камере, характеризуется положением в пространстве относительно сцены и внутренними параметрами. Внутренние параметры наблюдателя полностью совпадают с таковыми у камеры, а положение наблюдателя определяет преобразование из его СК в мировую. Таким образом, преобразование, задаваемое внешними параметрами камеры, обратно преобразованию, определяемому внешними параметрами наблюдателя. Пусть матрица поворота R_c и вектор переноса t_c описывают положение камеры, а матрица поворота R_v и вектор переноса t_v — положение наблюдателя, тогда

$$R_c = R_v^{-1}; \quad T_c = -R_v^{-1}t_v.$$

3 Виртуальная съемка

Модуль Viz предоставляет возможность интерактивного перемещения наблюдателя относительно сцены с помощью устройств ввода [5], что позволяет имитировать процесс съемки сцены. В процессе виртуальной съемки оператор позиционирует камеру и делает снимки с необходимых ракурсов. В качестве сцены в этом случае выступает полигональная модель. В простейшем случае полигональная модель состоит из упорядоченного множества вершин и множества граней, каждая из которых задана списком составляющих ее вершин. Каждая грань должна быть плоской, поэтому наибольшее распространение получили мо-

дели, гранями которых являются треугольники, так как три точки, не лежащие на одной прямой, однозначно определяют плоскость. Существуют несколько распространенных форматов хранения трехмерных моделей. Остановимся на формате PLY, так как он поддерживается библиотекой OpenCV. В сети Интернет присутствуют хранилища PLY-моделей, полученных как путем сканирования реальных объектов [6–8], так и путем моделирования в графических средах [9].

Каждый снимок характеризуется параметрами камеры и наблюдаемыми на этом снимке точками сцены с координатами их проекций. Для того чтобы указать, что проекции на разных снимках соответствуют одной и той же точке сцены, используется уникальный дескриптор. Если при формировании снимка учитывать только вершины модели, то в качестве дескриптора можно рассматривать номер вершины. Дескриптор точки вместе с координатами проекции будем называть наблюдением. Точки, наблюдаемые только на одном снимке, не подлежат реконструкции; следовательно, они не должны присутствовать в выходных данных виртуальной съемки. Возможна ситуация, когда в процессе съемки не будут охвачены все вершины полигональной модели, что повлечет за собой непоследовательность дескрипторов и необходимость их перенумерации. С учетом вышеизложенного можно заключить, что запись данных снимка в файл непосредственно при его взятии невозможна. Вместо этого будем сохранять их в некоторой структуре данных, а по окончании виртуальной съемки исключим точки, наблюдаемые только с одного снимка, и перенумеруем оставшиеся, приведя дескрипторы к диапазону $[0; n]$, где n — число наблюдаемых точек.

Выделение точек, не подлежащих реконструкции, и перенумерация дескрипторов являются тривиальными задачами и могут быть реализованы разными методами. Если использовать сортировку подсчетом, то вычислительная сложность составит $O(N)$ по времени и памяти, где N — число вершин полигональной модели. Если применить сортировку, основанную на сравнениях, то можно добиться сложности $O(n \log n)$ по времени, где n — общее число наблюдений.

Возможность изменения внутренних параметров камеры можно обеспечить, добавив дополнительные элементы графического интерфейса, позволяющие изменять фокусное расстояние и коэффициенты дисторсии. Установить внутренние параметры камеры для окна визуализации можно с помощью функции `Viz3d::setCamera(...)`, а получить текущие параметры — `Viz3d::getCamera(...)`. Внешние параметры камеры можно узнать, обратив матрицу аффинного преобразования, возвращаемую функцией `Viz3d::getViewerPose(...)`, и выделив в полученном преобразовании составляющую поворота и составляющую переноса с помощью функций `Affine3d::rotation()` и `Affine3d::translation()`.

Использование высокоуровневого модуля Viz накладывает ряд ограничений, в числе которых невозможность отображения модели с учетом радиальных искажений; значит, дисторсии могут быть использованы только в расчетах.

Библиотека OpenCV предоставляет возможность сохранения структур данных в файлы форматов XML и YAML, что можно использовать для сохранения результата виртуальной съемки.

4 Реалистичность набора данных

Будем обеспечивать реалистичность полученного набора данных, основываясь на трех признаках: отсутствие невидимых точек, неточность локализации проекций, наличие ложных соответствий.

Невидимые точки можно разделить на две группы: точки, попавшие за пределы поля зрения камеры, и точки, закрытые видимой частью сцены. Точка попадает в поле зрения камеры, если она лежит спереди камеры и координаты ее проекции не выходят за границы снимка. Пусть точка (X, Y, Z) имеет координаты (X', Y', Z') в СК, связанной с камерой, и проецируется в точку (x, y) на снимке. Тогда точка попадает в поле зрения камеры, если $Z' > 0$, $0 \leq x \leq W$ и $0 \leq y \leq H$, где W и H — ширина и высота снимка соответственно. Если $Z' \leq 0$, то координаты проекции можно не вычислять — точка заведомо находится за пределами поля зрения.

Точки, попадающие в поле зрения камеры, могут быть перекрыты другими частями сцены. Геометрически условие видимости точки можно сформулировать так: точка будет видима тогда и только тогда, когда отрезок, соединяющий эту точку с центром проекции, не имеет общих точек со сценой, кроме той, которая является его концом.

Задача определения видимых частей сцены решается графическим адаптером с помощью Z -буферизации во время рендеринга модели [10]. В Z -буфере для каждого пикселя полученного изображения модели закодирована координата Z в СК, связанной с камерой. Значит, для определения видимости некоторой точки необходимо сравнить ее Z -координату со значением в Z -буфере для соответствующего ей пикселя. На уровне модуля Viz отсутствует доступ к Z -буферу, поэтому модуль Viz был модифицирован добавлением функции `Viz3d::getDepth(...)`, позволяющей обратиться к любой ячейке буфера глубины. Библиотека OpenCV с модифицированным модулем Viz находится в открытом репозитории¹.

Неточность в координатах проекций является неотъемлемым признаком реальных снимков и возникает ввиду дискретизации изображения. Простой способ внести ошибку — округлить координаты до целых, иными словами, заменить координаты точки координатами ближайшего к ней пикселя. Недостаток такого варианта — невозможность регулировать степень неточности координат. Другой способ, получивший значительное признание в области компьютерного зрения, состоит в добавлении к координатам нормально распределенной случайной величины $N(0, \sigma^2)$. Увеличивая значение среднеквадратического отклонения σ , можно увеличивать разброс координат относительно их истинных значений, что позволяет исследовать устойчивость алгоритмов реконструкции к величине ошибки в координатах проекций.

Ни один из существующих алгоритмов выделения особых точек и поиска соответствий между ними не может гарантировать отсутствие ложных соответствий.

¹<https://github.com/helgui/opencv>.

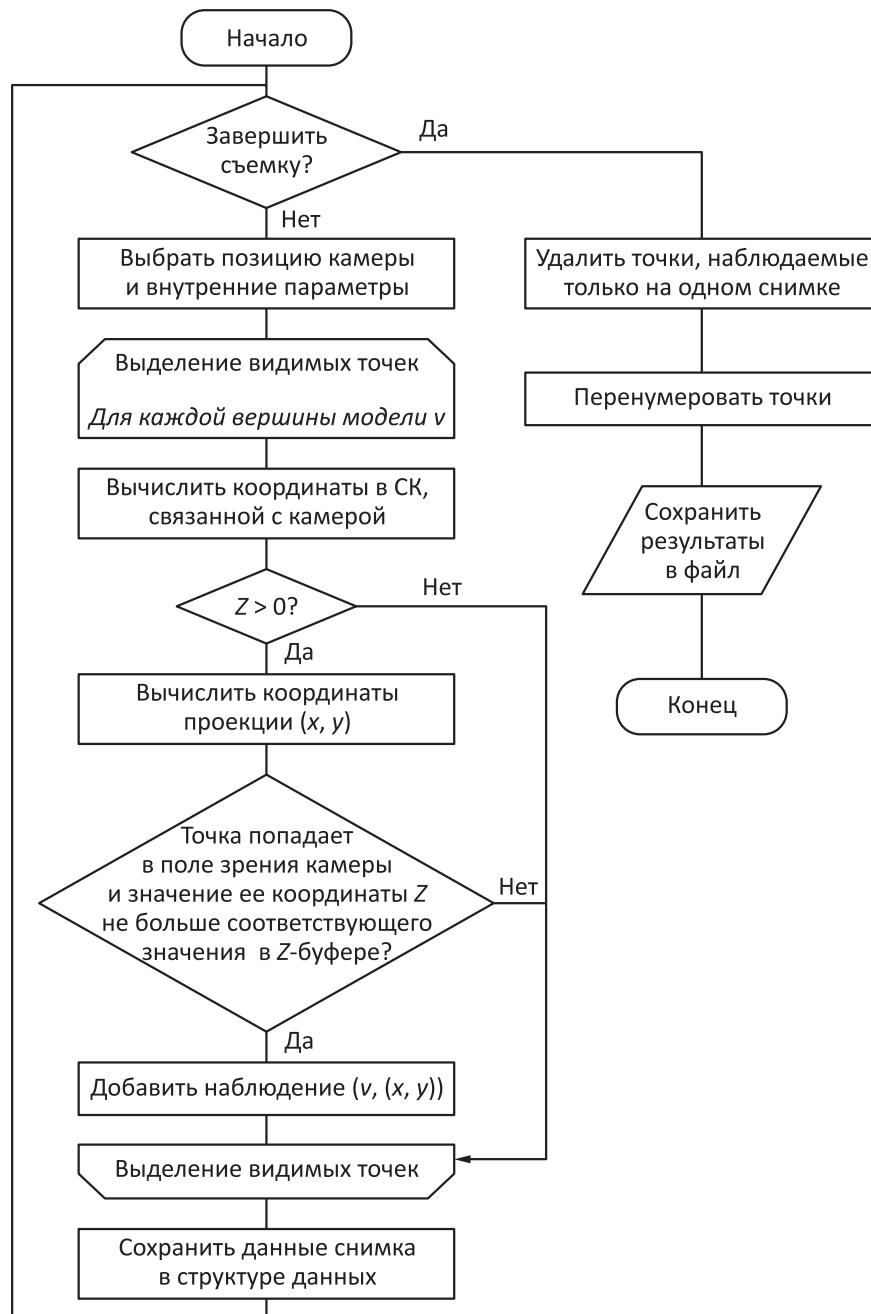


Рис. 1 Блок-схема процесса виртуальной съемки

Добавление ложных соответствий в результаты виртуальной съемки тривиально, важно лишь предусмотреть возможность варьировать их абсолютное или относительное количество.

Внесение ошибки в координаты проекций и добавление ложных соответствий должны быть отделены от процесса виртуальной съемки, чтобы упростить создание наборов данных, отличающихся лишь параметрами реалистичности. Такой подход также значительно упрощает добавление других критериев реалистичности и соответствующих им операций. На рис. 1 изображена блок-схема процесса виртуальной съемки с учетом вышеизложенных процедур.

5 Заключение

Рассмотрен метод виртуальной съемки, позволяющий создавать наборы данных для алгоритмов трехмерной реконструкции, отмечены особенности его реализации с использованием библиотеки OpenCV. В соответствии с изложенным материалом было разработано ПО, исходный код которого на языке C++ размещен в открытом репозитории¹.

Разработанное ПО позволяет создавать наборы данных в интерактивном режиме, осуществлять их просмотр, изменять наборы данных в соответствии с критериями реалистичности. Необходимая процедура и ее параметры указываются в строке запуска программы. Для создания набора данных используется команда `gen`, для просмотра созданного набора — команда `view`, для внесения погрешности в координаты проекций — команда `noisp`, для добавления ложных соответствий — команда `falsm`. Параметры каждой из процедур подробно рассмотрены в справочном файле, находящемся в репозитории с исходным кодом. Программа поддерживает файлы форматов PLY (3D-модель), YAML, XML и TXT (наборы данных).

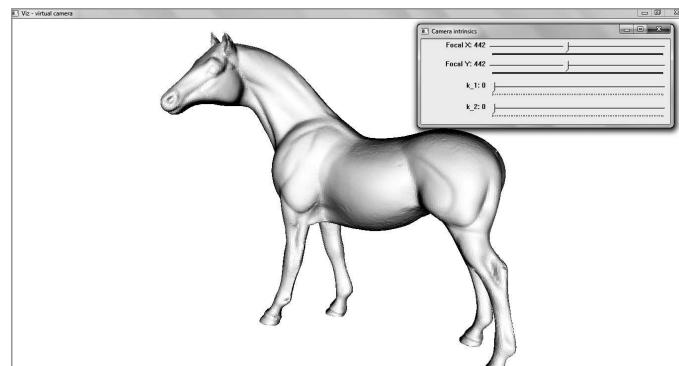


Рис. 2 Виртуальная съемка

¹<https://github.com/helgui/SfmDataGenerator>.

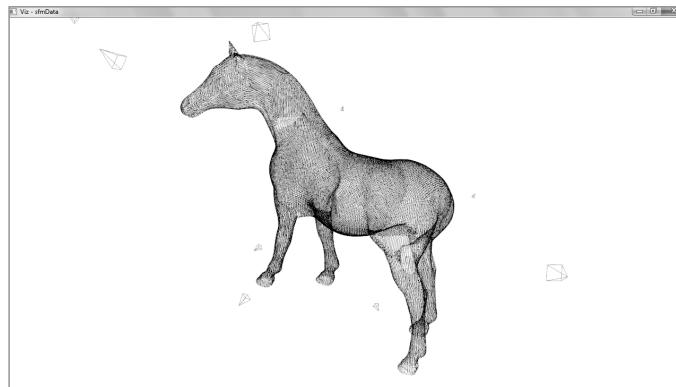


Рис. 3 Визуализация набора данных

При создании набора данных оператору доступно окно виртуальной съемки и окно изменения внутренних параметров камеры (рис. 2). Позиционирование камеры осуществляется с помощью мыши и клавиатуры в соответствии с документацией модуля Viz [5]. Для фиксации снимка используется клавиша «пробел». Для завершения виртуальной съемки достаточно закрыть окно визуализации или нажать клавишу «Q».

При просмотре набора данных оператору отображаются позиции, из которых были сделаны снимки, и облако точек (рис. 3). Управление просмотром осуществляется так же, как и при создании набора данных.

В качестве дальнейшего развития работы рассматривается возможность визуализации модели с учетом дисторсий, разработка метода оценки качества реконструкции, инвариантного к преобразованию масштабирования.

Литература

1. Hartley R., Zisserman A. Multiple view geometry in computer vision. — 2nd ed. — Cambridge: Cambridge University Press, 2003. 670 p.
2. Moulon P., Monasse P., Marlet R. Adaptive structure from motion with a contrario model estimation // 11th Asian Conference on Computer Vision Proceedings. — Springer, 2012. Part IV. P. 257–270.
3. Furukawa Y., Hernandez C. Multi-view stereo: A tutorial // Foundations Trends Computer Graphics Vision, 2013. Vol. 9. No. 1-2. P. 1–148.
4. Seitz S. M., Curless B., Diebel J., Scharstein D., Szeliski R. A Comparison and evaluation of multi-view stereo reconstruction algorithms // 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) Proceedings. — IEEE, 2006. Vol. 1. P. 519–528.
5. OpenCV: 3D visualizer. http://docs.opencv.org/3.1.0/d1/d19/group_viz.html#gsc.tab=0.

6. The Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep>.
7. Large geometric models archive. http://www.cc.gatech.edu/projects/large_models.
8. KSCAN3D gallery. <http://www.kscan3d.com/gallery>.
9. PLY Files: An ASCII Polygon format. <http://people.sc.fsu.edu/~jburkardt/data/ply/ply.html>.
10. Foley J. D., van Dam A., Feiner S. K., Hughes J. F. Computer graphics: Principles and practice. — Boston: Addison Wesley, 1997. 1264 p.

Поступила в редакцию 15.03.16

GENERATING REALISTIC STRUCTURE-FROM-MOTION DATASETS THROUGH VIRTUAL PHOTOGRAPHY

O. A. Yakovlev and A. V. Gasilov

Orel Branch of the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 137 Moskovskoe Shosse, Orel 302025, Russian Federation

Abstract: The paper presents the virtual photography method of generating structure-from-motion datasets. Virtual photography is an interactive process of making shots from different viewpoints and with arbitrary camera intrinsics according to the pinhole camera model. The result of virtual photography may be considered as input data for the structure-from-motion software. The paper provides realistic datasets according to the most significant properties of real datasets. Implementation of the virtual photography method using the OpenCV library is described.

Keywords: structure from motion; synthetic datasets; virtual shooting; OpenCV

DOI: 10.14357/08696527160206

References

1. Hartley, R., and A. Zisserman. 2004. *Multiple view geometry in computer vision*. 2nd ed. Cambridge: Cambridge University Press. 670 p.
2. Moulon, P., P. Monasse, and R. Marlet. 2012. Adaptive structure from motion with a contrario model estimation. *11th Asian Conference on Computer Vision Proceedings*. 4:257–270.
3. Furukawa, Y., and C. Hernandez. 2013. Multi-view stereo: A tutorial. *Foundations Trends Computer Graphics Vision* 9(1-2):1–148.
4. Seitz, S. M., B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings*. 1:519–528.

5. OpenCV: 3D visualizer. Available at: http://docs.opencv.org/3.1.0/d1/d19/group_viz.html#gsc.tab=0 (accessed March 9, 2016).
6. The Stanford 3D Scanning Repository. Available at: <http://graphics.stanford.edu/data/3Dscanrep/> (accessed March 9, 2016).
7. Large geometric models archive. Available at: http://www.cc.gatech.edu/projects/large_models/ (accessed March 9, 2016).
8. KSCAN3D gallery. Available at: <http://www.kscan3d.com/gallery/> (accessed March 9, 2016).
9. PLY Files: An ASCII Polygon format. Available at: <http://people.sc.fsu.edu/~jburkardt/data/ply/pl.html> (accessed March 9, 2016).
10. Foley, J. D., A. van Dam, S. K. Feiner, and J. F. Hughes. 1997. *Computer graphics: Principles and practice*. Boston: Addison Wesley. 1264 p.

Received March 15, 2016

Contributors

Yakovlev Oleg A. (b. 1992) — research engineer, Orel Branch of the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences; 137 Moskovskoe Shosse, Orel 302025, Russian Federation; maucra@gmail.com

Gasilov Artur V. (b. 1992) — research engineer, Orel Branch of the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences; 137 Moskovskoe Hwy., Orel 302025, Russian Federation; gasilov.av@ya.ru

ИЕРАРХИЧЕСКАЯ ФОРМА ПРЕДСТАВЛЕНИЯ БИОГРАФИЧЕСКОГО ФАКТА

И. М. Адамович¹, О. И. Волков²

Аннотация: Данная статья посвящена факту в конкретно-историческом исследовании. Показана специфика биографического факта (БФ), проявляющаяся в его двойственности, отражающей его связи как с реальным миром, так и с информационным пространством (ИП). Описана иерархическая форма представления БФ, использующаяся в системе Т-парсер для автоматического извлечения фактов из текстов. Доказано соответствие данного представления формальным свойствам БФ. Определены и описаны операции логики фактов для данного представления. Также продемонстрирована связь иерархического представления фактов с онтологией предметной области и вытекающая из этого перспективность ее использования в автоматизированных процедурах обработки фактов. Показана возможность частичной формализации фактов на этапе их выделения. Описана технология обработки данных в биографическом исследовании, автоматизирующая как этап сбора информации, так и этап сопоставления выявленных фактов с целью взаимоувязывания и разрешения противоречий, включающая полную формализацию фактов на базе онтологии. Показаны проблемы этого подхода и намечены пути их решения.

Ключевые слова: биографический факт; иерархическая форма факта; автоматизированная технология обработки фактов; онтология; логика фактов

DOI: 10.14357/08696527160207

1 Введение

Специфика биографического исследования состоит в том, что в центре внимания исследователя находится конкретная личность и все без исключения стороны (социальные, экономические, политические, этнические, художественные и т. п.) ее реальной жизни [1]. Соответственно БФ, выявляемые в результате исследования, отличаются большим многообразием как по составу характеристик, так и по связям между ними.

Уже разработаны и используются средства автоматизации обработки БФ (сравнения, нормализации, выявления связей, разрешения противоречий, вывода новых фактов). Примером может служить разработанная в ИПИ ФИЦ ИУ

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, Adam@amsd.com

²Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, Volkov@amsd.com

РАН система «Фактограф» [2], обеспечивающая формализацию, хранение, интеграцию, аналитико-синтетическую обработку БФ, получаемых из гетерогенных источников. Но выявление БФ остается очень трудоемким процессом и отнимает у исследователя много сил и времени [3], поэтому очень востребованными являются системы автоматического извлечения фактов из текстов на естественном языке (ЕЯ), в том числе биографической и художественно-исторической направленности.

Работы по интеллектуальной обработке ЕЯ-текстов и извлечению из них структур знаний давно ведутся в лаборатории Компьютерной лингвистики и когнитивных технологий обработки текстов ИПИ ФИЦ ИУ РАН. В основе этих работ лежит созданная И. П. Кузнецовым концепция формализма представления знаний, получившего название «расширенные семантические сети» (РСС), а также разработанный им язык ДЕКЛ, обеспечивающий гибкое преобразование структур РСС, включая поиск и порождение новых знаний [4–8]. В рамках дальнейшего развития этого подхода была разработана система Т-парсер [9] для извлечения структурированных данных из текста при помощи контекстно-свободных грамматик. Т-парсер опирается на иерархическое (древовидное) представление факта. Принципиальная адекватность такого представления сути БФ была наглядно показана в [9], но, вообще говоря, утверждение о строгом соответствии данного представления формальным признакам БФ требует строгого обоснования, что и является основной целью данной статьи.

Также показана связь данной структуры с онтологией предметной области и вытекающая из этого перспективность ее использования в автоматизированных процедурах обработки.

2 Специфика биографического факта

Существуют различные определения факта в научном смысле этого слова (научного факта). Но все они сводятся к тому, что научный факт отражает объективные свойства реальных объектов и процессов. Даже в областях, где невозможно непосредственное взаимодействие с объектом (например, в палеонтологии), исследователь имеет дело с объективными свидетельствами (окаменелостями, ископаемыми останками). Совсем другое дело — факт в конкретно-историческом исследовании (частным случаем которого является биографическое исследование). Исследователь, как правило, имеет дело с субъективными свидетельствами и оценками. Даже в ситуации, когда, казалось бы, можно опереться на объективные данные, невозможно исключить субъективную составляющую. Например, информацию о росте персоны — объекта исследования можно получить объективным измерением, если сохранилась его одежда. Но факт принадлежности этой одежды искомой персоне устанавливается исключительно по семейным преданиям, чьим-то свидетельствам и т. п., т. е. на основании субъективной информации.

Информация, зафиксированная документально, образует ИП — бумажные и другие твердые носители, а в последние десятилетия и электронные ресур-

сы [10]. Информационное пространство формирует образ реального мира, в котором реальные объекты (РО) имеют свое отражение. Информационный образ (ИО) при этом не тождественен РО. Следовательно, факты, содержащиеся в документах, адекватно описывают образы РО, но не сами объекты. Более того, один РО может иметь несколько ИО в ИП. Цель же исследователя — установление научных фактов, получение объективной информации о РО исследования (насколько это вообще возможно в научном исследовании). Поэтому биограф работает с фактами двух типов:

- (1) факты, описывающие ИО (*i*-факты). Обладают свойством фрагментарности и противоречивости. Не существуют в отрыве от своих метаданных, т. е. информации о документе-источнике.
Получение этих фактов — необходимая и наиболее трудоемкая стадия биографического исследования, состоящая в работе в архивах и просмотре огромного числа документов. Автоматизация этого этапа является важной и актуальной задачей в связи со все увеличивающимся общественным интересом к семейной истории;
- (2) факты, описывающие РО (*r*-факты), т. е. объективное знание об объекте исследования. Работа исследователя на втором этапе исследования состоит в сопоставлении *i*-фактов между собой, а также с ранее выявленными *r*-фактами, относящимися к объекту исследования, и с нормаллями (наборами правил, регламентирующих зависимости между фактами) [11] с целью взаимоувязывания и разрешения противоречий. При этом *r*-факты после публикации результатов исследования могут выступать *i*-фактами в каком-либо другом исследовании.

3 Иерархическая форма представления биографического факта

Формы представления информации могут быть самыми различными, но не все они подходят к БФ в силу их специфики: так, семантическая сеть, представляющая собой узлы, соответствующие объектам предметной области, и дуги, связывающие узлы и описывающие отношения между ними, вполне пригодна для хранения информации [12], но *i*-факты принципиально фрагментарны, а отношения, задаваемые ими, нечетки, неоднозначны и взаимно противоречивы. Так, при изучении даже таких официальных документов, как метрические книги и данные переписи населения, часто приходится сталкиваться с разным указанием даты рождения одного и того же человека в разных документах. Даже степень родства может быть указана по-разному. Что же касается такой информации, как причина смерти в разделе «Об умерших», то ее достоверность крайне низка, поскольку она устанавливалась без участия медицинских работников.

Образовать четкую сеть могут только *r*-факты, что, собственно, и является целью исследования, но это их свойство никак не может быть использовано в процессе его проведения.

Также существуют достаточно универсальные формы представления, в принципе подходящие и для БФ. К примеру, в Томита-парсере компании Яндекс — средстве для извлечения структурированных данных из текста на ЕЯ [13] — используется табличная форма представления фактов. Такая форма позволяет эффективно описывать структуру факта, хотя и требует несколько громоздкого механизма объединения таблиц для сведения фактов в факты более высокого уровня, необходимость в чем обусловливается многопроходной технологией извлечения информации. В случае же биографического исследования сама структура извлекаемого из источника факта не всегда может быть описана заранее. Следовательно, структура БФ должна быть адаптирована:

- к специфике биографической информации;
- к технологии поиска фактов;
- к технологии дальнейшей работы с ними с учетом возможности их автоматизированной обработки.

В [3] предложена иерархическая форма представления факта и показано ее соответствие специфике биографической информации и удобство ее использования для описанной технологии автоматического извлечения фактов из биографических и художественно-исторических текстов с помощью системы Т-парсер. Также было показано принципиальное соответствие данной структуры самому понятию БФ. Так, в соответствии с [10], под БФ понимается утверждение, что некая характеристика некоторого объекта принимает некоторое конкретное значение. Применительно к Т-парсеру характеристикой, подразумевающей наличие своего объекта, является правило, а значением — выделенный этим правилом фрагмент разбираемого предложения. Так, правило «фамилия», выделившее в тексте

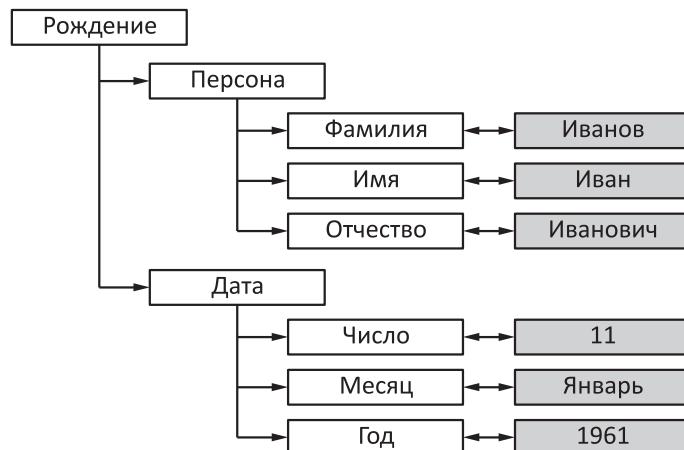


Рис. 1 Пример факта «рождение»

фрагмент «Иванов», сформирует факт, что некто имеет фамилию Иванов, т. е. существует объект, у которого характеристика «фамилия» принимает значение «Иванов».

В соответствии с логикой работы Т-парсера результат применения правила, в свою очередь, может быть выделен неким правилом, что формирует иерархию фактов, т. е. под фактом в Т-парсере понимается иерархическая структура именованных понятий, терминальные узлы которой содержат выделенные из фразы слова или словосочетания, которые интерпретируются как их значения (рис. 1).

Вообще говоря, для иерархического представления факта не является принципиальным, что значения имеют только терминальные узлы. Для общности следует считать, что значения есть у каждого узла и значением по умолчанию является факт его наличия. Так, если в приведенном примере убрать значение «Иванович» у узла «Отчество», то это следует интерпретировать как факт наличия отчества у персоны.

4 Соответствие основным свойствам биографического факта

Как показано в [10], БФ должен подчиняться правилам логики фактов, т. е. для БФ должны быть определены конъюнкция, дизъюнкция, импликация и отрицание. Также для БФ должны быть справедливы основные свойства: симметрия, транзитивность и вариативность. Покажем, что представление факта как иерархической структуры не противоречит этим требованиям.

Конечно же, существует тривиальная трактовка этого требования: любой факт, независимо от формы его представления, можно трактовать как высказывание, могущее быть как истинным, так и ложным (*i*-факт вполне может быть ложным, если объект ИП, который он описывает, неадекватно отражает соответствующий объект реального мира), а поэтому к ним применимы все логические операции булевой алгебры. Но такой подход не является конструктивным для БФ, поскольку истинность *i*-факта заранее не известна и ее только требуется установить в процессе исследования. Поэтому следует рассмотреть более конструктивные, специфические для выбранной формы представления фактов, аналоги основных логических операций.

4.1 Формальное описание факта

Введем ряд обозначений и понятий. Для этого опишем иерархический факт Tree как совокупность узлов, представляющих собой совокупность именованного понятия и значения, которое, вообще говоря, может быть пусто, рекуррентной формулой

$$\text{Tree} = \{\text{Node}, \{\text{Tree}_i | i = 1, \dots, m\}\},$$

т. е. дерево представляет собой корневой узел и множество его поддеревьев в количестве m штук. Для терминальных узлов множество поддеревьев пусто, т. е. $m = 0$. Корневой узел любого (в том числе являющегося частью другого



Рис. 2 Два эквивалентных дерева: (a) Tree, (б) $\text{Tree}' = \text{Eq}(\text{Tree})_{\text{Node}_2}$

дерева) дерева будем обозначать как Rt , т. е. $\text{Node} = \text{Rt}(\text{Tree})$. Обратную операцию обозначим как Tr , т. е. $\text{Tree} \equiv \text{Tr}(\text{Rt}(\text{Tree}))$.

Для любого узла Node_n , удаленного от корневого на расстояние n , существует последовательность $\{\text{Node}_0, \dots, \text{Node}_{n-1}\}$ вышестоящих узлов, где Node_0 — корневой узел. Тогда иерархический факт можно описать формулой:

$$\begin{aligned} \text{Tree} = & \{\text{Node}_0, \{\text{Tree}_i^0 | \text{Rt}(\text{Tree}_i^0) \neq \text{Node}_1\}, \\ & \text{Node}_1, \{\text{Tree}_i^1 | \text{Rt}(\text{Tree}_i^1) \neq \text{Node}_2\}, \dots \\ & \dots, \text{Node}_{n-1}, \{\text{Tree}_i^{n-1} | \text{Rt}(\text{Tree}_i^{n-1}) \neq \text{Node}_n\}, \text{Node}_n, \{\text{Tree}_i^n\}\}. \end{aligned}$$

Будем говорить, что дерево Tree' эквивалентно дереву Tree относительно узла Node_n , если

$$\begin{aligned} \text{Tree}' = & \{\text{Node}_n, \{\text{Tree}_i^n\}, \text{Node}_{n-1}, \{\text{Tree}_i^{n-1} | \text{Rt}(\text{Tree}_i^{n-1}) \neq \text{Node}_n\}, \dots \\ & \dots, \text{Node}_1, \{\text{Tree}_i^1 | \text{Rt}(\text{Tree}_i^1) \neq \text{Node}_2\}, \\ & \text{Node}_0, \{\text{Tree}_i^0 | \text{Rt}(\text{Tree}_i^0) \neq \text{Node}_1\}\}, \end{aligned}$$

т. е. состав узлов в Tree' совпадает с Tree , порядок узлов для цепочки $\{\text{Node}_0, \dots, \text{Node}_n\}$ меняется на обратный, а для остальных сохраняется. Корневым узлом в Tree' становится Node_n (рис. 2). Кратко будем записывать этот факт как $\text{Tree}' = \text{Eq}(\text{Tree})_{\text{Node}_n}$.

Далее факты будем обозначать заглавными латинскими буквами. Для i -фактов, не существующих в отрыве от своей метаинформации, вводим соответствующее обозначение: факт A с метаинформацией M будем обозначать как $A|M$.

4.2 Операции над фактами

Теперь рассмотрим аналоги основных логических операций применительно к БФ в иерархической форме, для которых введем соответствующие обозначения:

1. F -конъюнкция ($\&_F$).

Определим F -конъюнкцию следующим образом:

$$\exists N_1 \in A|_{M_1}, N_2 \in B|_{M_2} : \text{Tr}(N_1) = \text{Tr}(N_2) \Rightarrow A|_{M_1} \&_F B|_{M_2} = \text{Tr}(N_1),$$

т. е. операция определена и имеет смысл только для i -фактов, причем с различной метаинформацией, что соответствует фактам, полученным из разных источников. Если эти факты имеют совпадающее поддерево (т. е. общий подфакт), то этот подфакт можно считать имеющим независимое подтверждение из разных источников и, следовательно, достоверным. Соответственно, данный подфакт можно считать новым r -фактом, являющимся результатом F -конъюнкции исходных i -фактов. Таким образом, операцию F -конъюнкции можно назвать операцией сверки.

2. F -дизъюнкция (\vee_F).

Определим F -дизъюнкцию следующим образом:

$$\exists N_1 \in A, N_2 \in B : \text{Tr}(N_1) = \text{Tr}(N_2) \Rightarrow A \vee_F B = \text{Eq}(A)_{N_1} \cup \text{Eq}(B)_{N_2}.$$

Операция представляет собой объединение двух фактов в один. Следовательно, она определена и имеет смысл только для фактов с одинаковой метаинформацией, что соответствует фактам, полученным из одного источника, или для r -фактов. Факты приводятся к эквивалентному виду относительно корневых узлов общего поддерева и объединяются в единое дерево.

3. F -импликация (\Rightarrow_F).

Определим F -импликацию следующим образом:

$$A \Rightarrow_F B = B|_A.$$

Это неформальная операция — один из важных элементов этапа обработки фактов в биографическом исследовании. Как показано в [3], поиск информации в биографическом исследовании проходит итерационно и направление поиска на последующих этапах зависит от результатов поиска на предыдущем этапе. Так, операцию выявления нового направления в исследовании задает именно F -импликация, т. е. формирование важного с точки зрения исследования факта, логически зависящего (как правило, на основании некоторых нормалей) от истинности некоторого i -факта, который сам по себе может казаться малозначимым. Так, факт службы в армии некоторой персоны в определенное время может означать невозможность его участия в это время в некотором важном событии. С позиций описываемой структуры это соответствует новому i -факту, в качестве метаданных которого принимается этот малозначимый факт.

4. F -отрицание (\neg_N).

F -отрицание является обычным булевским отрицанием и поэтому не требует специального значка F в обозначении, но, поскольку применяется к конкретному узлу факта, обозначение включает узел. Так, $B = \neg_N A$ означает, что факт B совпадает с фактом A , кроме значения узла N , которое заменено на его логическое отрицание.

4.3 Свойства фактов

В соответствии с [10] БФ должен обладать следующими свойствами:

1. Симметрия.

Для большинства двуместных отношений факту $\beta(p, q)$, где p и q — объекты, однозначно соответствует факт $\beta'(q, p)$.

Пример: *МестоРаботы*(«Иванов», «Контора») \Leftrightarrow *Сотрудник*(«Контора», «Иванов»).

Данное свойство для БФ в иерархическом представлении безусловно выполняется. Так, если β есть факт с иерархией (*ФИО* = «Иванов») \rightarrow (*Работа* = «Контора»), то $\beta' = \text{Eq}(\beta)_{\text{Работа}}$.

2. Транзитивность.

Факты, основанные на таких характеристиках, как иерархия, обладают свойством транзитивности. Пример: *Работа* («Иванов», «Контора») \wedge *Место* («Контора», «Москва») \Rightarrow *Место* («Иванов», «Москва»).

Иерархическое представление БФ автоматически поддерживает свойства, основанные на иерархии понятий. Так, из иерархии понятий, приведенных на рис. 3, выделяются цепочки:

- a) Иванов–Контора: (*Фамилия* = «Иванов») \leftarrow (*Персона*) \rightarrow (*Работа*) \rightarrow (*Название* = «Контора»);
- б) Иванов–Москва: (*Фамилия* = «Иванов») \leftarrow (*Персона*) \rightarrow (*Работа*) \rightarrow (*Место* = «Москва»).

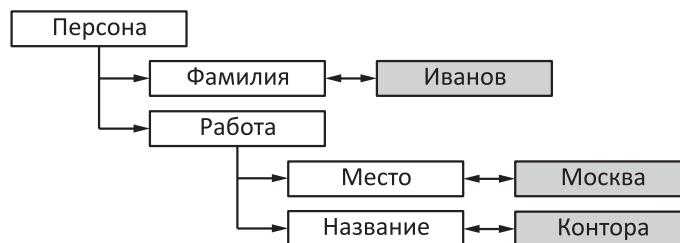


Рис. 3 Пример иерархии понятий

3. Вариативность. Для вариативных характеристик, например именования, значения разбиваются на классы эквивалентности, определяемые нормальными, а формула (речь идет о «нормализованной» записи факта $(\forall t \in \Delta t)\beta(p, q, t) = a$, означающей, что некая характеристика объекта или объектов в некий момент времени принимает определенное значение) принимает вид принадлежности данному классу.

Сама по себе иерархическая форма представления факта не накладывает никаких дополнительных ограничений на форму представления значений характеристик, но, как будет показано далее, легко может быть увязана с онтологией, необходимой для дальнейшей обработки фактов, что решает проблему вариативности за счет формализации понятий предметной области.

5 Связь иерархической формы факта с онтологией

Специфика *i*-фактов такова, что их превращение в *r*-факты, т. е. в факты в научном понимании этого слова, представляет собой весьма нетривиальную и не формализуемую на 100% процедуру. Но такая ее важная составляющая, как процедура сопоставления *i*-фактов с целью выявления информационных лакун и противоречий, вполне может быть автоматизирована. Существует специализированное программное обеспечение, разработанное специально для сопоставления биографической информации [2]. Но для любой автоматизированной процедуры обработки данные (т. е. факты) должны быть формализованы. А поскольку БФ формируются из текстов на ЕЯ, для успешного определения значений слов и правильных связей между ними необходимо знание мира, который описывается в тексте. Такое знание предоставляется онтологией. Онтология представляет собой формализацию знаний о взаимосвязях объектов и целых классов объектов реального мира, которая позволяет компьютеру использовать эти знания и даже дополнять информацию об отдельных объектах с помощью логического вывода [14]. Подходы к задачам выделения фактов из текстов на ЕЯ с опорой на онтологии активно разрабатываются в последнее время [15–17]. Более того, одно из определений факта звучит как «содержащаяся в тексте и метаданных документа совокупность связей между сущностями, описываемыми в онтологии информационной системы» [18].

Частичная формализация на основе онтологии составляющих БФ в иерархическом представлении вполне возможна и вытекает из технологии формирования таких фактов. Так, на рис. 4 схематически представлена технология формирования БФ, позволяющая применять автоматизацию как на этапе формирования *i*-фактов, так и на этапе формирования *r*-фактов с использованием онтологии. Технология предполагает выполнение следующих этапов:

1. Формируются правила определения фактов. Имена правил привязываются к онтологии.

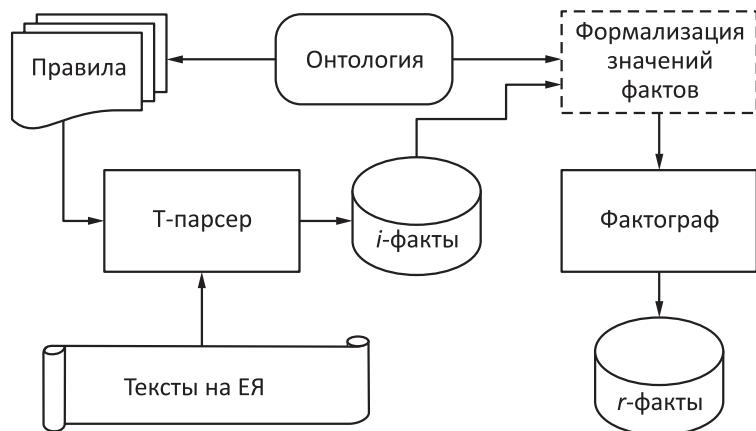


Рис. 4 Автоматизированная технология формирования *r*-фактов

2. Посредством Т-парсера с использованием подготовленных правил из текстов на ЕЯ вычленяются *i*-факты в иерархической форме. Имена узлов иерархии при этом формализованы за счет привязки извлекающих правил к онтологии. Тем самым, поскольку имена характеристик для большинства узлов и определяют значения типа «факт наличия», часть значений становится формализованной уже на этом этапе.
3. Формализуются непустые значения узлов. В данный момент это осуществляется вручную, поскольку применение «Фактографа» осуществлялось для обработки информации, полученной из уже частично формализованных источников — метрических книг и исповедных ведомостей. В дальнейшем при использовании большого объема данных, полученных из текстов на ЕЯ, потребуется автоматизация и этого процесса.
4. С использованием модуля «Фактограф» формируются *r*-факты. В перспективе возможно также использование иных модулей, отражающих специфику изучаемых объектов и предметной области.

6 Выводы

Описано и обосновано иерархическое представление БФ. Описана автоматизированная технология формирования БФ из текстов на ЕЯ. Намечены пути развития данной технологии. При этом следует отметить и проблемы, присущие предложенному подходу:

- сложность создания онтологии с учетом широты предметной области биографического исследования. В качестве решения этой проблемы могут быть предложены следующие подходы:

- (а) децентрализация создания онтологии [19];
- (б) автоматизация создания онтологии [20];
- отсутствие автоматизированной процедуры формализации непустых значений узлов. Данная проблема непосредственно связана с проблемой создания онтологии и может быть легко решена при наличии таковой. Видится целесообразным объединение процедуры формализации на основе онтологии с процедурой парсинга текста на ЕЯ, т. е. дооснащение Т-парсера средствами взаимодействия с онтологией.

Поскольку элементы описанной технологии уже применяются и показали свою эффективность для ряда задач биографического поиска, следует считать выбранную форму представления БФ приемлемой, а предложенную технологию — перспективной.

Литература

1. *Иконникова С. Н.* Биографика как часть исторической культурологии // Вестник СПбГУКИ, 2012. № 2(11). С. 6–10.
2. *Адамович И. М.* Методы и средства справочно-поисковой поддержки научных и социально-культурных проектов на основе интеграции данных разнородных биографических источников: Отчет о НИР. — М.: ИПИ РАН, 2012. С. 67–84.
3. *Адамович И. М., Волков О. И.* Средства поддержки интернет-поиска при проведении биографических исследований // Системы и средства информатики, 2014. Т. 24. № 2. С. 178–192.
4. *Kuznetsov I. P., Kozerenko E. B., Kuznetsov K. I., Timonina N. O.* Intelligent System for Entities Extraction (ISEE) from natural language texts // Workshop (International) on Conceptual Structures for Extracting Natural Language Semantics (SENSE'09) at the 17th Conference (International) on Conceptual Structures (ICCS'09) Proceedings / Eds. U. Priss, G. Angelova. — Moscow, Russia: University Higher School of Economics, 2009. P. 17–25.
5. *Kuznetsov I. P., Kozerenko E. B., Matskevich A. G.* Intelligent extraction of knowledge structures from natural language texts // 2011 IEEE/WIC/ACM Joint Conferences (International) on Web Intelligence and Intelligent Agent Technology (WI-IAT 2011) Proceedings, 2011. P. 269–272.
6. *Kozerenko E. B., Ermakov P. V.* The strategies of syntactic analysis based on head-driven grammars and the methods of their implementation in information systems // Информатика и её применения, 2011. Т. 5. Вып. 4. С. 107–113.
7. *Кузнецов И. П., Сомин Н. В.* Выявление имплицитной информации из текстов на естественном языке: проблемы и методы // Информатика и её применения, 2012. Т. 6. Вып. 1. С. 49–58.
8. *Шарнин М. М., Кузнецов И. П.* Особенности семантического поиска информационных объектов на основе технологии баз знаний // Информатика и её применения, 2012. Т. 6. Вып. 2. С. 113–121.
9. *Адамович И. М., Волков О. И.* Система извлечения биографических фактов из текстов исторической направленности // Системы и средства информатики, 2015. Вып. 25. № 3. С. 235–250.

10. *Маркова Н. А.* Логика биографических фактов // Информатика и ее применения, 2012. Т. 6. Вып. 2. С. 87–96.
11. *Маркова Н. А.* Электронная коллекция биографических фактов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Тр. XIV Всеросс. науч. конф. RCDL'2012. — Переславль-Залесский: Университет города Переславля, 2012. С. 287–293.
12. *Мизарбеков А. М.* Алгоритм хранения информации в семантической сети // Известия ЮФУ. Технические науки, 2012. Т. 130. Вып. 5. С. 25–28.
13. *Огородник Р. В., Серебреная Л. В.* Обработка текста с помощью Томита-парсера // Информационные технологии и системы 2014 (ИТС-2014): Мат-лы междунар. науч. конф. — Минск: БГУИР, 2014. С. 230–231.
14. *Диконов В. Г., Богуславский И. М., Тимошенко С. П.* Онтология для поддержки задач извлечения смысла из текста на естественном языке // Информационные технологии и системы (ИТиС'12): Сб. тр. 35-й Конф. молодых ученых и специалистов ИППИ РАН. — М.: ИППИ РАН, 2012. С. 152–161.
15. *Сидорова Е. А.* Инструментальные средства фактографического анализа документов в информационных системах, основанных на онтологии // Вестник НГУ. Сер. Информационные технологии, 2008. Т. 6. Вып. 3. С. 126–134.
16. *Святогор Л. А., Гладун В. П.* Машинальное понимание текстов естественного языка: онтологическая парадигма // Искусственный интеллект, 2010. № 3. С. 249–258.
17. *Оробинская Е. А., Дорошенко А. Ю.* Использование онтологий для автоматической обработки текстов на естественном языке // Вестник НТУ ХПИ: Сб. научных тр., 2011. № 30. С. 101–106.
18. *Барахнин В. Б., Федотов А. М.* Построение модели фактографического поиска // Вестник НГУ. Сер. Информационные технологии, 2013. Т. 11. Вып. 4. С. 16–27.
19. *Слободюк А. А., Маторин С. И., Четвериков С. Н.* О подходе к созданию онтологий на основе системно-объектных моделей предметной области // Научные ведомости БелГУ, 2013. № 22(165). Вып. 28 / 1. С. 159–167.
20. *Найханова Л. В.* Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования. — Улан-Удэ: БНЦ СО РАН, 2008. 244 с.

Поступила в редакцию 18.12.15

HIERARCHICAL FORMAT OF A BIOGRAPHICAL FACT

I. M. Adamovich and O. I. Volkov

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: This article focuses on the facts in the specific historical studies. The specific duality of a biographical fact, which is evidenced in its connections with

the real world and with the information space, is mentioned. The hierarchical format of a biographical fact, which is used in the “T-parser” system of automatic extraction of facts from texts in natural language, is described in detail. The accordance of this format with the formal properties of a biographical fact is proven. The logic of biographical facts operators for this format is explored and described. The relation of the hierarchical format of facts with the domain ontology and the prospects of its usage in automated systems of facts processing are also demonstrated. The possibility of partial formalization at the stage of fact extraction is analyzed. The technology of biographical data processing, which automates the fact extraction stage and the stage of facts comparison with the purpose of integration and conflict resolution, including full fact formalization on the basis of an ontology, is proposed and analyzed. The problems of this approach are described and the possible ways of their solution are proposed.

Keywords: biographical fact; hierarchical format of fact; automated technology of facts processing; ontology; logic of facts

DOI: 10.14357/08696527160207

References

1. Ikonomikova, S. N. 2012. Biografika kak chast' istoricheskoy kul'turologii [Biographical studies as part of the historical cultural studies]. *Vestnik SPbGU* [Bulletin of Saint-Petersburg State University of Culture and Art] 2(11):6–10.
2. Adamovich, I. M. 2012. Metody i sredstva spravochno-poiskovoy podderzhki nauchnykh i sotsial'no-kul'turnykh proektorov na osnove integratsii dannykh raznorodnykh biograficheskikh istochnikov [Methods and tools of information support of scientific and sociocultural projects based on the integration of heterogeneous data of different biographical sources]. Research Report. Moscow: IPI FRC CSC RAS. 67–84.
3. Adamovich, I. M., and O. I. Volkov. 2014. Sredstva podderzhki internet-poiska pri provedenii biograficheskikh issledovanii [The technology of Internet-searching as the part of the biographic investigation]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 2(24):178–192.
4. Kuznetsov, I. P., E. B. Kozerenko, K. I. Kuznetsov, and N. O. Timonina. 2009. Intelligent System for Entities Extraction (ISEE) from natural language texts. *Workshop (International) on Conceptual Structures for Extracting Natural Language Semantics (SENSE'09) at the 17th Conference (International) on Conceptual Structures (ICCS'09) Proceedings*. Eds. U. Priss and G. Angelova. Moscow, Russia: University Higher School of Economics. 17–25.
5. Kuznetsov, I. P., E. B. Kozerenko, and A. Matskevich. 2011. Intelligent extraction of knowledge structures from natural language texts. *2011 IEEE/WIC/ACM Joint Conferences (International) on Web Intelligence and Intelligent Agent Technology (WI-IAT 2011) Proceedings*. 269–272.
6. Kozerenko, E. B., and P. V. Ermakov. 2011. The strategies of syntactic analysis based on head-driven grammars and the methods of their implementation in information systems. *Informatika i ee Primeneniya — Inform. Appl.* 5(4):107–113.
7. Kuznetsov, I. P., and N. V. Somin. 2012. Vyayavlenie implitsitnoy informatsii iz tekstov na estestvennom yazyke: Problemy i metody [Extraction of implicit information from natural language: Problems and methods].

- the texts in natural language: Problems and methods]. *Informatika i ee Primeneniya — Inform. Appl.* 6(1):49–58.
8. Sharnin, M. M., and I. P. Kuznetsov. 2012. Osobennosti semanticheskogo poiska informatsionnykh ob'ektov na osnove tekhnologii baz znaniy [Semantic search of natural language information on the basis of knowledge base technology]. *Informatika i ee Primeneniya — Inform. Appl.* 6(2):113–121.
 9. Adamovich, I. M., and O. I. Volkov. 2015. Sistema izvlecheniya biograficheskikh faktov iz tekstov istoricheskoy napravленности [The system of facts extraction from historical texts]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 25(3):235–250.
 10. Markova, N. A. 2012. Logika biograficheskikh faktov [A logic of biographical facts]. *Informatika i ee Primeneniya — Inform. Appl.* 6(2):49–58.
 11. Markova, N. A. 2012. Elektronnaya kolleksiya biograficheskikh faktov [Digital collection of biographic facts]. *14th All-Russian Scientific Conference “Digital Libraries: Advanced Methods and Technologies, Digital Collections” Proceedings*. Pereslavl-Zalesky, Russia. 287–293.
 12. Mirzabekov, Ya. M. 2012. Algoritm khraneniya informatsii v semanticheskoy seti [Algorithm for storing information in semantic networks]. *Izvestiya UFU. Tekhnicheskie nauki* [Herald of SFedU. Engineering Sciences] 130(5):25–28.
 13. Ogorodnik, R. V., and L. V. Serebrenaya. 2014. Obrabotka teksta s pomoshch'yu Tomita-parsera [Text processing using Tomita-parser]. *Scientific Symposium (International) “Information Technology and Systems 2014” Proceedings*. Minsk. 220–231.
 14. Dikonorov, V. G., I. M. Boguslavsky, and S. P. Timoshenko. 2013. Ontologiya dlya podderzhki zadach izvlecheniya smysla iz teksta na estestvennom yazyke [Ontology to support semantic analysis of natural language texts]. *35th Conference “Information Technologies and Systems” (ITiS’12) Proceedings*. Moscow: IPPI RAN. 152–161.
 15. Sidorova, E. A. 2008. Instrumental'nye sredstva faktograficheskogo analiza dokumentov v informatsionnykh sistemakh, osnovannykh na ontologiyakh [Factographic text analysis tools in information systems based on ontologies]. *Novosibirsk State University J. Information Technologies* 6(3):126–134.
 16. Sviatogor, L. A., and V. P. Gladun. 2010. Mashinnoe ponimanie tekstov estestvennogo yazyka: Ontologicheskaya paradigma [Machine understanding of natural language texts: An ontological paradigm]. *Artificial Intelligence* 3:249–258.
 17. Orobinska, E. A., and A. Yu. Doroshenko. 2011. Ispol'zovanie ontologiy dlya avtomaticheskoy obrabotki tekstov na estestvennom yazyke [Ontologies using for natural language texts automatic processing]. *Sb. nauch. tr. “Vestnik NTU “KhPI”* [NTU “KhPI” Proceedings]. Kharkov. 30:101–106.
 18. Barakhnin, V. B., and A. M. Fedotov. 2013. Postroenie modeli faktograficheskogo poiska [A model of factographic retrieval]. *Novosibirsk State University J. Information Technologies* 11(4):16–27.
 19. Slobodyuk, A. A., S. I. Matorin, and S. N. Chetverikov. 2013. O podkhode k sozdaniyu ontologiy na osnove sistemno-ob'ektnykh modeley predmetnoy oblasti [About approach for building ontologies based on UFO domain models]. *Nauchnye Vedomosti BelGU* [Belgorod State University Scientific Bulletin] 22(28/1):159–167.
 20. Naikhanova, L. V. 2008. *Tekhnologiya sozdaniya metodov avtomaticheskogo postroeniya ontologiy s primenением генетического и автоматного программирования* [The

technology of automatic ontology building with genetic and automate programming methods creation]. Ulan-Ude: The Buryat Scientific Center of SB RAS. 244 p.

Received December 18, 2015

Contributors

Adamovich Igor M. (b. 1934) — Candidate of Science (PhD) in technology, Head of Department, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Adam@amsd.com

Volkov Oleg I. (b. 1964) — leading programmer, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; Volkov@amsd.com

ИНФОРМАЦИОННАЯ СИСТЕМА ЯЧЕЕЧНО-НЕЙРОСЕТЕВОГО МОДЕЛИРОВАНИЯ ПОСЛЕДСТВИЙ ХИМИЧЕСКИХ АВАРИЙ НА ОПАСНЫХ ПРОИЗВОДСТВЕННЫХ ОБЪЕКТАХ

С. П. Дударов¹, П. Л. Папаев²

Аннотация: Рассмотрена информационная система (ИС), предназначенная для моделирования процесса рассеяния примеси загрязняющего вещества в атмосферном воздухе в результате аварийного выброса на опасном производственном объекте. Одновременно с ее помощью могут быть решены задачи оценки последствий выброса через расчет зон загрязнения и зон токсического поражения. В качестве математического средства моделирования в системе используются ячеично-нейросетевые модели, комбинирующие преимущества искусственных нейронных сетей и ячеекных моделей. Архитектура разработанной ИС представлена с помощью организационной и функциональной структур, а также информационно-логической модели базы данных. Помимо описаны этапы работы с программно-алгоритмическим обеспечением системы: обработка выборок данных, обучение нейронной сети, работа с базовыми и обобщающими моделями.

Ключевые слова: ячеично-нейросетевое моделирование; экологическая безопасность; последствия химической аварии; концентрация загрязняющего вещества; поле концентрации; информационно-моделирующая система; программно-алгоритмическое обеспечение; организационная структура; функциональная структура; база данных; информационно-логическая модель

DOI: 10.14357/08696527160208

1 Введение

Программно-алгоритмическое обеспечение в области экологической и промышленной безопасности отличается многообразием благодаря большому числу решаемых задач и используемых в них моделей, методов и методик. К ним относятся, в частности, логико-графические и вероятностные модели анализа и оценки риска, модели и методики рассеяния загрязняющих веществ в результате аварийного или постоянно действующего выброса, методики оценки экологического и социального ущерба [1]. Вследствие сложных вычислительных расчетов, неизменно необходимых для получения, например, полей концентраций загрязняющих веществ и динамики ее изменения под воздействием внешних

¹Российский химико-технологический университет им. Д. И. Менделеева, dudarov@hotbox.ru

²Российский химико-технологический университет им. Д. И. Менделеева, papaev.p.l@inbox.ru

факторов, каждая модель или методика оценки последствий выбросов должна быть обеспечена соответствующим программным инструментарием.

В свою очередь, современный уровень развития информационных компьютерных технологий, характеризующийся не только высокими вычислительными мощностями, но и выдающимися возможностями средств графической визуализации, хранения и обработки данных, предъявляет особые требования к организации и функциональным возможностям информационно-моделирующих систем [2], разрабатываемых в том числе на основе специального математического и методического обеспечения в области экологической безопасности. Именно с учетом таких требований была разработана ИС ячеично-нейросетевого моделирования, рассмотренная в данной статье. Ее основное назначение — моделирование процесса рассеяния примеси загрязняющего вещества в атмосферном воздухе в результате аварийного выброса на опасном производственном объекте и, на основе результатов моделирования, оценка экологических и социальных последствий произошедшей аварии.

2 Ячеично-нейросетевые модели в задачах экологической безопасности

Подробная классификация ячеично-нейросетевых моделей, принципы их работы и примеры использования для решения прикладных задач в области экологической безопасности изложены в работах [3, 4]. Для понимания объемов работы и сложности задач, решавшихся при реализации описываемой информационно-моделирующей системы, отметим следующее.

Ячеично-нейросетевые модели, как следует из названия, — это синтез ячеичных моделей и искусственных нейронных сетей, а точнее — однослойных и многослойных перцептронов [3]. Территория, подвергающаяся загрязнению, разбивается на участки (ячейки) одинаковой формы такого размера, при котором с допустимой погрешностью можно считать, что переменные объекта моделирования (атмосферного воздуха) и внешней среды неизменны в любой точке ячейки. Процесс рассеяния примеси при этом определяется ее перераспределением между соседними ячейками в каждый дискретный момент времени с выбранным шагом. В качестве математического описания этого процесса выступают заранее обученные однослойные или многослойные перцептроны.

К сожалению, для решения задачи моделирования рассеяния примеси загрязняющего вещества невозможно собрать достаточный объем практических наблюдений, чтобы хорошо обучить перцепtron. Поэтому обучающая выборка формируется на основе вычислительного эксперимента, полученного по другим, стандартным расчетным методикам, например [5]. Недостаток этих методик — возможность рассчитывать поле концентрации загрязняющего вещества исключительно для постоянных метеорологических условий и условий местности, причем как на короткие (до 1 ч), так и на продолжительные (десятки часов) интервалы времени. Благодаря временной и пространственной дискретизации полей

концентрации и использованию в одной обучающей выборке примеров, полученных при различных исходных данных, в ячеично-нейросетевых моделях можно задавать сценарии изменения условий внешней среды и, таким образом, получать поля концентрации, в большей степени соответствующие реальной картине, и, как следствие, точнее рассчитывать экологические и социальные последствия химических аварий.

3 Организационная структура информационно-моделирующей системы

В данной работе рассматриваются вопросы создания программно-алгоритмического обеспечения для моделирования последствий химических аварий на опасных производственных объектах с использованием ячеично-нейросетевых моделей, предложенных в статьях [3, 4]. Оно реализовано в интегрированной среде разработки приложений Delphi 7. С его помощью можно получить концентрации загрязняющего вещества в точке и/или некоторой локальной окрестности, профили концентрации во времени, поля концентрации загрязняющего вещества, профили концентрации вдоль осей координат и профили токсодоз для любой точки прогнозируемого поля концентрации. Общая постановка задачи, которая ставилась перед разработкой данной системы, подробно описана в работе [4].

Организационная структура любой ИС предполагает наличие самостоятельных блоков, имеющих собственную, не всегда одинаковую программную или технологическую платформу. Также она предполагает участие человека. Это могут быть различные группы пользователей: администратор, оператор, руководитель, — принимающие решение каждый на своем уровне, в пределах своей компетенции. Блоки организационной структуры и пользователи системы связываются между собой информационными и управляющими потоками данных. Разработанная ИС ячеично-нейросетевого моделирования последствий химических аварий на опасных химических объектах организована в соответствии со структурой, приведенной на рис. 1.

В состав организационной структуры входят:

- подсистема технических средств, включающая вычислительные средства, а также средства ввода данных и вывода (представления) информации пользователю, базирующиеся на современных информационно-коммуникационных технологиях;
- подсистема программных средств, включающая стандартное и специализированное программно-алгоритмическое обеспечение, взаимодействие с которым персонала осуществляется через интерфейс пользователя системы.

Стандартное программное обеспечение, в свою очередь, включает современные операционную систему, систему управления базами данных и средства для работы с электронными документами и таблицами. Специализированное



Рис. 1 Организационная структура ИС

программное обеспечение — это комплекс математического и программенно-алгоритмического обеспечения, в котором реализована вся методическая база и учтена специфика решаемых задач экологической безопасности атмосферы промышленных территорий.

4 Функциональная структура информационно-моделирующей системы

Функциональная структура любой ИС включает в себя комплекс подсистем, каждая из которых имеет свое собственное, достаточно четко ограниченное назначение. В основном подсистемы реализуются в виде самостоятельных модулей программно-алгоритмического обеспечения, служащих для осуществления человека-машинного взаимодействия с техническими средствами и базой данных системы. Само человеко-машинное взаимодействие обусловливается моделями и методами, реализованными в программном комплексе.

На рис. 2 представлена функциональная структура разработанной информационно-моделирующей системы. В ней, исходя из их функционального на-

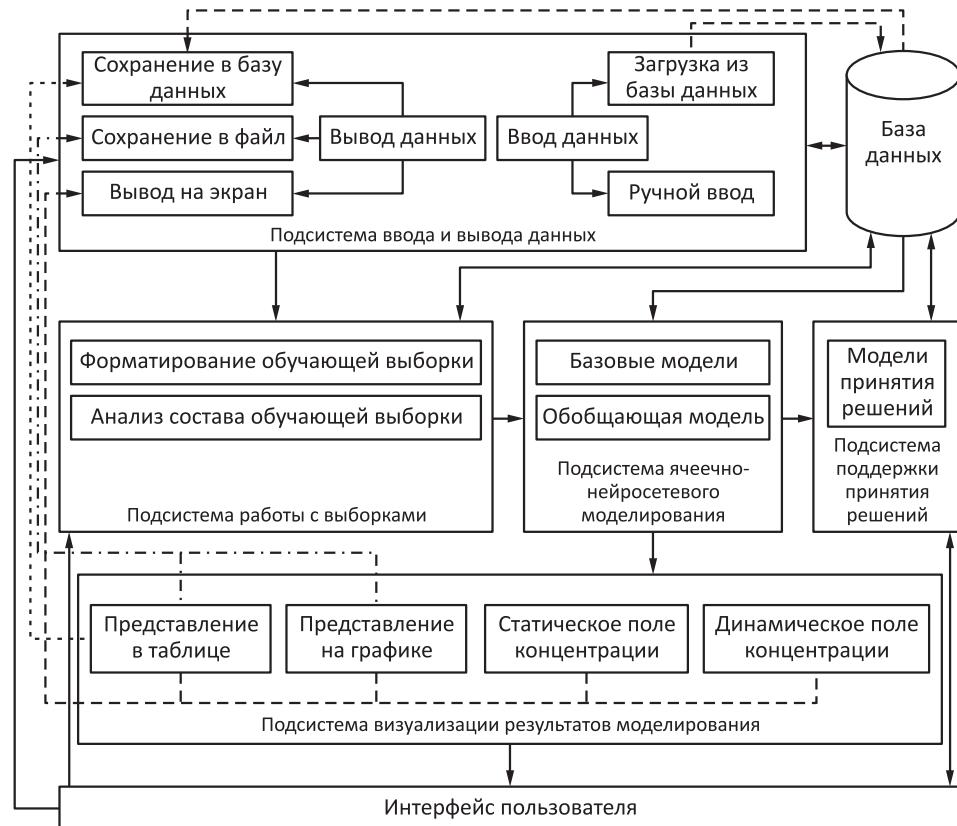


Рис. 2 Функциональная структура ИС

значения, выделяются пять основных подсистем. За основу взята, переработана и дополнена функциональная структура, ранее описанная в работе [6].

Подсистема ввода и вывода данных предназначена для загрузки и выгрузки универсальной и обучающей выборок, использующихся для обучения нейросетевых моделей, ввода и сохранения настроек ячеично-нейросетевых моделей, исходных данных по условиям моделирования, сохранения результатов моделирования в файлах данных и отчетов. Источником и целью назначения указанной информации в зависимости от ее характера и использующей подсистемы являются рабочие файлы системы, файлы отчетов и база данных.

В подсистеме работы с выборками реализован полный перечень функций для формирования и анализа состава обучающих выборок: проверка наличия и исключение повторяющихся и противоречивых примеров, нормализация и де-нормализация, масштабирование, анализ репрезентативности, корреляционный анализ.

Подсистема ячеично-нейросетевого моделирования позволяет получать модели переноса или распределения загрязняющей примеси между ячейками на основе обученных нейронных сетей, а также производить расчеты с использованием полученных моделей.

Подсистема поддержки принятия решений реализует соответствующую функцию на основе моделей принятия решений по управлению экологической безопасностью, заложенных в информационно-моделирующей системе.

Благодаря подсистеме визуализации результатов моделирования на экран монитора пользователю выводятся все результаты моделирования в удобной для него форме: графической, табличной, текстовой.

5 База данных информационно-моделирующей системы

Большой объем данных, используемых в системе для обучения ячеично-нейросетевых моделей, их настройки, хранения и других целей обусловил необходимость разработки базы данных. Для ее реализации используется система управления базами данных MySQL. Структура связей данных и их состав приведен в информационно-логической модели на рис. 3.

Таблица «Универсальные выборки» содержит описание, а также характеристики источника выброса и метеорологических условий, при которых были получены примеры, входящие в состав таких выборок. Самые примеры располагаются в отдельной одноименной таблице. Каждая запись содержит ссылочный ключ на универсальную выборку, к которой относится данный пример. Состав полей таблицы «Примеры выборки» таков, что в ней можно хранить обучающие и тестовые данные для любых видов окрестностей ячеично-нейросетевых моделей. Поля, не используемые для окрестностей целевого вида, остаются пустыми.

Теоретически в базе данных могут содержаться примеры, полученные как в ходе натурного, так и в ходе вычислительного эксперимента. На практике, в связи с отсутствием натурных экспериментальных данных для каких-либо видов окрестностей ячеично-нейросетевых моделей, в базе хранятся и используются для обучения нейронных сетей только примеры, полученные по методике оценки последствий химических аварий [5].

При формировании выборок для отдельных нейросетевых моделей из универсальной выборки импортируются необходимые данные. В таблице «Нейросетевые модели» для каждой переменной указывается ее вид (входная или выходная) и, при помощи ссылочного ключа на таблицу «Свойства переменных», способ масштабирования. Поле также остается пустым, если переменная не задействуется в окрестности модели. Также в таблице «Нейросетевые модели» хранятся все настройки нейронной сети и алгоритма ее обучения.

В результате удовлетворительного обучения нейронной сети полученные весовые коэффициенты сохраняются в одноименную таблицу.

Все сохраненные в базе данных нейросетевые модели могут применяться в обобщающей модели, которая используется для формирования полей концен-

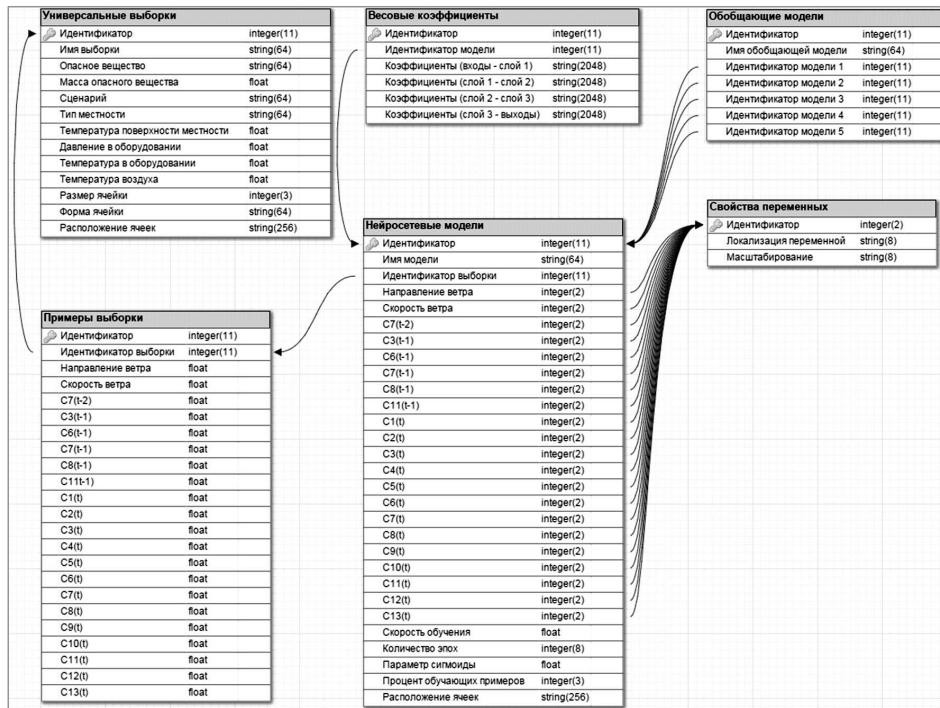


Рис. 3 Информационно-логическая модель базы данных информационно-моделирующей системы

трации в различные моменты времени после аварийного выброса химического вещества. Это становится возможным благодаря тому, что каждая модель решает свою узкоспециализированную расчетную задачу: интерполяция и экстраполирование концентрации примеси в ячейках, расчет продольных и поперечных профилей, прогнозирование концентрации на заданный шаг по времени. Идентификаторы базовых моделей (не менее двух и не более пяти), входящих в одну обобщающую модель, перечисляются в таблице «Обобщающие модели».

6 Реализация программно-алгоритмического обеспечения

Разработанное программно-алгоритмическое обеспечение имеет стандартный графический, интуитивно понятный, дружественный пользователю интерфейс. При реализации ИС использовались стандартные элементы интерфейса windows-приложений (рис. 4): главное меню, панель вкладок, панель инструментов для каждой из вкладок и рабочее поле программы, организованное с использованием стандартных элементов интерфейса (таблиц, графиков, полей ввода, кнопок управления и др.).

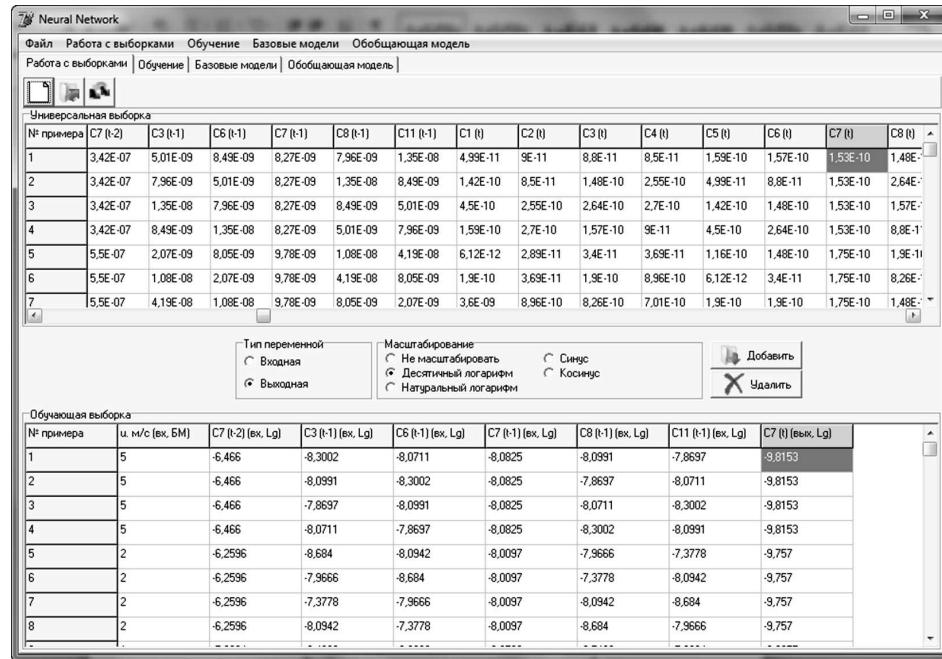


Рис. 4 Пример экранной формы программно-алгоритмического обеспечения информационно-моделирующей системы

При решении задачи ячеично-нейросетевого моделирования загрязнения воздуха в результате аварийного выброса работу в программе можно разделить на 4 этапа:

- (1) работа с выборками данных;
- (2) обучение нейронной сети;
- (3) работа с базовыми моделями;
- (4) работа с обобщающей моделью.

Информационная система построена таким образом, что можно начать работу в ней с любого этапа, предварительно загрузив необходимые данные из базы данных и / или файла.

На первом этапе, проиллюстрированном на рис. 4, из универсальной выборки формируются обучающая и тестовая выборки с указанием свойств переменных для последующего обучения нейронной сети. Для начала работы необходимо загрузить универсальную выборку из базы данных. При этом сформируется таблица «Универсальная выборка» в рабочем поле программы. Далее можно приступить к формированию выборки примеров под ту или иную модель, указы-

вая вид переменной (входная или выходная) и способ ее масштабирования, если это необходимо.

Второй этап — обучение нейронной сети. Импортировав обучающую выборку из раздела «Работа с выборками» или из базы данных, необходимо задать настройки нейронной сети: указать ее структуру (количество скрытых слоев и нейронов в каждом из них; число входов и выходов считается автоматически), настройки нейронов (вид и параметр активационной функции), настройки алгоритма обучения (количество эпох, коэффициент скорости). После этого можно приступить непосредственно к процессу обучения, нажав соответствующую управляющую кнопку. Для оптимизации весовых коэффициентов в системе использован алгоритм обратного распространения ошибки. Рассчитанные весовые коэффициенты и настройки сети сохраняются в базу данных.

Каждая обученная нейронная сеть соответствует определенной базовой ячеично-нейросетевой модели. На третьем этапе реализована работа с ранее сохраненными моделями. Рабочее поле программы в этом разделе делится на три фрейма: настройки модели, входные данные и результаты (рис. 5). Содержимое фреймов варьируется в зависимости от выбранной модели и решаемой локальной задачи.

На последнем этапе, когда уже получены все базовые модели, реализована работа с обобщающей моделью. С ее помощью можно рассчитать и построить

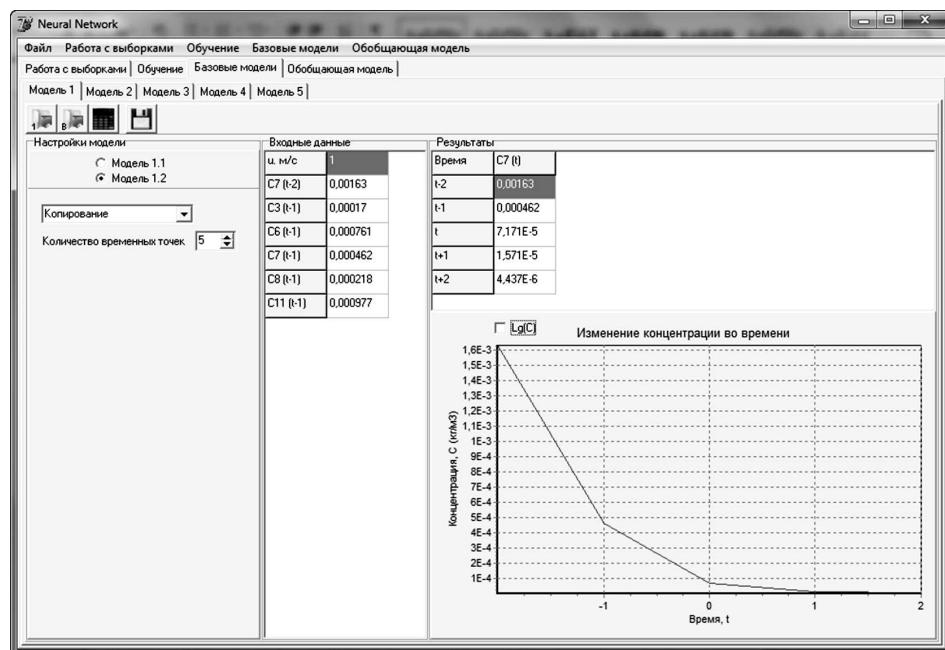


Рис. 5 Пример работы с готовой ячеечно-нейросетевой моделью

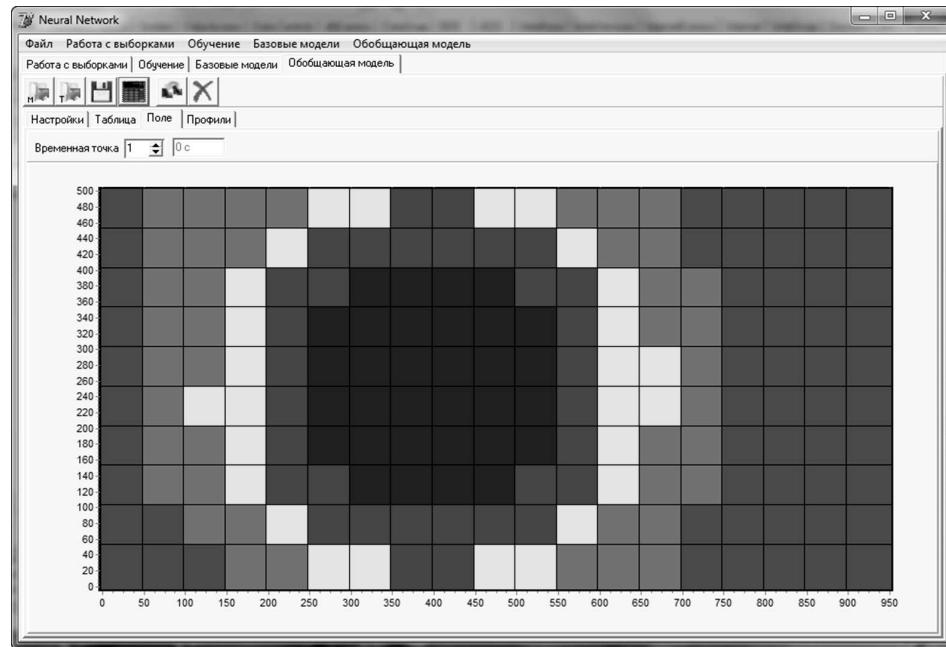


Рис. 6 Интерфейс экранной формы «Обобщающая модель» (вкладка «Поле»)

профили и поля концентрации и токсодоз загрязняющего вещества в табличной и графической формах в различные моменты времени. Данный этап (рис. 6) предусматривает ввод настроек обобщающей модели (вкладка «Настройки»), ввод значений концентрации примеси в определенных точках в известные моменты времени (вкладка «Таблица») и получение результатов расчета на вкладках «Поле» и «Профили».

В ходе настройки обобщающей модели задаются размеры исследуемой территории, число временных отрезков и метеорологические характеристики (скорость и направление ветра). При указании характеристик ветра возможны два случая: постоянные и переменные метеоусловия. Существующие методики, включая [5], позволяют рассчитать поля концентрации и токсодоз только для постоянной комбинации метеоусловий. Пространственно-временная дискретизация в ячеично-нейросетевых моделях дает возможность получить результаты и для переменных метеоусловий. Соответствующие примеры приведены в работах [3, 4]. При выборе переменного типа метеоусловий пользователю необходимо ввести направление и скорость ветра, прогнозируемые для каждого отрезка времени.

Допускается не только сплошной, но и фрагментарный ввод исходных значений концентрации примеси. Координаты в заголовках столбцов и строк на вкладке «Таблица» определяют центр ячейки окрестности.

После ввода всех настроек обобщающей модели и исходных концентраций можно перейти к расчету. Алгоритм, реализованный в программном обеспечении информационно-моделирующей системы автоматически, в итерационном режиме подберет базовые модели для заполнения пустот в исходных данных. Соответствующие результаты отображаются на вкладках «Поле» и «Профили». Пример построенного поля концентрации приведен на рис. 6. На вкладке «Профили» отображаются графики токсодоз и концентрации загрязняющего вещества во времени и пространстве. Все возможные графики для любой координаты можно практически мгновенно получить, выбрав нужную ячейку на вкладке «Поле».

7 Заключение

Разработанная ИС была использована для решения задач моделирования последствий гипотетической аварии с выбросом газообразного аммиака [4] при постоянных и переменных метеоусловиях, а также для реально имевшего место выброса аммиака 13 июля 1973 г. в Потчефструме (ЮАР) [7]. Результаты моделирования с помощью системы подтвердились известными данными о жертвах рассмотренной аварии.

Описанная ИС ячеично-нейросетевого моделирования последствий химических аварий на опасных производственных объектах может использоваться для решения производственных, научных и учебных задач экологической безопасности атмосферы промышленных территорий. Также она может быть интересна в качестве прототипа при разработке ИС схожего назначения с собственным математическим и методическим обеспечением.

Перспективы развития описанной информационно-моделирующей системы связаны прежде всего с реализацией моделей, алгоритмов, пользовательского интерфейса и функций визуализации результатов работы подсистемы поддержки принятия решений.

Литература

1. Савицкая Т. В., Егоров А. Ф. Методы оценки рисков и негативных воздействий химически опасных объектов. — М.: РХТУ им. Д. И. Менделеева, 2011. 264 с.
2. Меньшиутина Н. В., Мамасов А. В. Современные информационные системы хранения, обработки и анализа данных для предприятий химической и смежных отраслей. — М.: РХТУ им. Д. И. Менделеева, 2011. 308 с.
3. Дударов С. П., Папаев П. Л., Кудряшов А. Н., Карабова Ю. А. Ячеично-нейросетевые модели в задачах экологической безопасности // Искусственный интеллект и принятие решений, 2011. № 2. С. 31–39.
4. Дударов С. П., Папаев П. Л., Колесов А. В. Оценка последствий химических аварий на опасных производственных объектах с использованием ячеично-нейросетевых моделей // Безопасность труда в промышленности, 2012. № 3. С. 64–70.
5. Иванов Е. А., Агапов А. А., Буйко К. В. и др. Методика оценки последствий химических аварий (Методика «Токси». Редакция 2.2) // Методики оценки последствий

- аварий на опасных производственных объектах: Сб. документов. — 3-е изд., испр. и доп. — М.: ЗАО НТИЦ ПБ, 2010. Сер. 27. Вып. 2. С. 123–206.
6. *Папаев П. Л., Дударов С. П.* Разработка организационной и функциональной структур информационно-моделирующей системы для оценки последствий аварий на химических производствах с использованием ячеично-нейросетевых моделей // Успехи в химии и химической технологии, 2013. Т. XXVII. № 1(141). С. 51–56.
 7. *Папаев П. Л., Дударов С. П.* Применение ячеично-нейросетевых моделей для оценки последствий залповых выбросов аммиака на примере реальной химической аварии // Успехи в химии и химической технологии, 2012. Т. XXVI. № 1(130). С. 29–33.

Поступила в редакцию 09.09.15

INFORMATION SYSTEM OF CELLULAR AND NEURAL NETWORK MODELING OF CHEMICAL ACCIDENT CONSEQUENCES AT DANGEROUS INDUSTRIAL FACILITIES

S. P. Dudarov and P. L. Papaev

D. I. Mendeleev University of Chemical Technology of Russia; 9 Miusskaya Sq.,
Moscow 125047, Russian Federation

Abstract: An information system intended for modeling the process of dissipation of contaminant in free air as a result of emergency emission at dangerous facility is described. The information system can also solve the task of emission consequences evaluation using pollution and toxic lesion areas calculation. Cellular and neural network models are applied as mathematical means of modeling. They combine advantages of both cellular models and artificial neural networks. The architecture of the developed information system is presented by organizational and functional structures as well as by the information-logical model of its database. The stages of the software use such as data samples processing, neural network training, work with the base and generalizing models are described thoroughly.

Keywords: cellular and neural network modeling; ecological safety; chemical accident consequences; contaminant concentration; concentration area; information system modeling; software development; information system organizational structure; information system functional structure; database; information-logical model

DOI: 10.14357/08696527160208

References

1. Savitskaya, T. V., and A. F. Egorov. 2011. Metody otsenki riskov i negativnykh vozdeystviy khimicheski opasnykh ob"ektov [Methods of an assessment of risks and negative impacts of chemically dangerous objects]. Moscow: D. I. Mendeleev University of Chemical Technology of Russia Publs. 264 p.

2. Menshutina, N. V., and A. V. Matasov. 2011. Sovremennoye informatsionnye sistemy khraneniya, obrabotki i analiza dannykh dlya predpriyatiy khimicheskoy i smezhnykh otrassley [Contemporary information systems of data storage, mining and analysis for enterprises of chemical and adjacent branches]. Moscow: D. I. Mendeleev University of Chemical Technology of Russia Publs. 308 p.
3. Dudarov, S. P., P. L. Papaev, A. N. Kudryashov, and Yu. A. Karibova. 2011. Yachechno-neyrosetevye modeli v zadachakh ekologicheskoy bezopasnosti [Cellular and neural network models in tasks of ecological safety]. *Iskusstvennyy Intellekt i Prinyatie Resheniy* [Artificial Intelligence and Decision Making] 2:31–39.
4. Dudarov, S. P., P. L. Papaev, and A. V. Kolosov. 2012. Otsenka posledstviy khimicheskikh avariiv na opasnykh proizvodstvennykh ob'ektakh s ispol'zovaniem yachechno-neyrosetevykh modeley [Consequences assessment of chemical accidents at hazardous production facilities using cell-neuron network models]. *Bezopasnost' Truda v Promyshlennosti* [Occupational Safety in Industry] 3:64–70.
5. Ivanov, E. A., A. A. Agapov, K. V. Buyko, et al. 2010. Metodika otsenki posledstviy khimicheskikh avariiv (Metodika "Toksi." Redaktsiya 2.2) [Technique of chemical accident consequence evaluation ("Toksi." technique. Edition 2.2)]. *Metodiki otsenki posledstviy avariiv na opasnykh proizvodstvennykh ob'ektakh: Sb. dokumentov* [Techniques of accident consequence evaluation at dangerous industrial objects. Document collection]. 3rd ed. Ser. 27. CJSC STC "Industrial Safety." 2:123–206.
6. Papaev, P. L., and S. P. Dudarov. 2013. Razrabotka organizatsionnoy i funktsional'noy struktur informatsionno-modeliruyushchey sistemy dlya otsenki posledstviy avariiv na khimicheskikh proizvodstvakh s ispol'zovaniem yachechno-neyrosetevykh modeley [Development of organizational and functional structures of information and modeling system for evaluation of accidents consequences at chemical facilities using cellular and neural network models]. *Uspekhi v Khimii i Khimicheskoy Tekhnologii* [Advances in Chemistry and Chemical Technology] 27(1):51–56.
7. Papaev, P. L., and S. P. Dudarov. 2012. Primenenie yachechno-neyrosetevykh modeley dlya otsenki posledstviy zalpovykh vybrosov ammiaka na primere real'noy khimicheskoy avariiv [Application of cellular and neural network models for evaluation of ammonia volley emission consequences on the example of real chemical accident]. *Uspekhi v Khimii i Khimicheskoy Tekhnologii* [Advances in Chemistry and Chemical Technology] 26(1):29–33.

Received September 9, 2015

Contributors

Dudarov Sergey P. (b. 1977) — Candidate of Science (PhD) in technology, associate professor, Dean of the Faculty of Information Technologies and Control, D. I. Mendeleev University of Chemical Technology of Russia; 9 Miusskaya Sq., Moscow 125047, Russian Federation; dudarov@hotbox.ru

Papaev Pavel L. (b. 1989) — PhD student, assistant professor, D. I. Mendeleev University of Chemical Technology of Russia; 9 Miusskaya Sq., Moscow 125047, Russian Federation; papaev.p.l@inbox.ru

**МЕТОДЫ ПОСТРОЕНИЯ СХЕМОТЕХНИЧЕСКИХ СИСТЕМ
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
С ИСПОЛЬЗОВАНИЕМ СЕРВИС-ОРИЕНТИРОВАННОГО
ПОДХОДА НА БАЗЕ ПРОТОКОЛА WEBSOCKET***

В. И. Анисимов¹, В. Н. Гридин², С. А. Васильев³

Аннотация: Приводятся основные критерии, которым должен соответствовать веб-сервис схемотехнического проектирования. Описывается классический подход к построению сервис-ориентированных систем. Раскрывается внутренняя архитектура веб-сервиса на основе сервис-ориентированного подхода (service-oriented architecture, SOA), а также основные этапы функционирования. Описывается применение SOA для систем автоматизированного проектирования (САПР), в частности схемотехнических САПР, и приводится пример успешного внедрения архитектуры SOA в САПР. Сформулирован и описан ряд схемотехнических задач для создания САПР на базе SOA. Рассматриваются преимущества и недостатки использования полнодуплексного протокола связи WebSocket (стандарт RFC 6455) для использования его в качестве основы ESB-шины сервис-ориентированной САПР относительно протокола SOAP. Приводятся преимущества и недостатки использования протокола WebSocket для организации внешних коммуникаций веб-сервиса. Освещается вопрос передачи сложных типов данных в гетерогенных средах программного обеспечения с применением полнодуплексного протокола данных WebSocket (стандарт RFC 6455). В заключении указываются преимущества и недостатки внедрения протокола связи WebSocket в САПР, в частности схемотехнического проектирования, на основе сервис-ориентированной архитектуры.

Ключевые слова: SOA; САПР; веб-сервис; веб-служба; WebSocket; среда .NET; шина ESB; кроссплатформенность; асинхронность; гетерогенная среда; частотная характеристика; стационарный режим; схемотехника; сложные типы данных; сетевой трафик; быстродействие

DOI: 10.14357/08696527160209

* Работа выполнена при финансовой поддержке Программы фундаментальных научных исследований ОНИТ РАН «Научные основы создания гетерогенных телекоммуникационных и локационных систем и их элементной базы».

¹ Центр информационных технологий в проектировании РАН (ЦИТП РАН); Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», vianisimov@inbox.ru

² Центр информационных технологий в проектировании РАН, info@ditc.ras.ru

³ Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», venom-gt@list.ru

1 Введение

В современном мире практически вся информация получает электронное воплощение, характеризуемое большими объемами данных, обработкой которых занимаются сервисы, отвечающие за преобразование, предоставление или хранение данных.

Как только сервис начинает представлять собой коммерчески выгодный продукт, стратегически важное государственное программное обеспечение или социально значимую службу, он начинает развиваться — увеличивается количество обращений или нагрузка в виде объема решаемых задач. В это время встает вопрос о надежности и отказоустойчивости, о стоимости, о доступности данного сервиса и, самое важное, о быстродействии, при этом под доступностью понимается удобство использования — удаленный доступ, а также поддержка кроссплатформенности клиентской стороны.

Действительно, универсального решения для создания быстрого и надежного, недорогого и доступного для широкой аудитории (удобного) веб-сервиса не существует.

Применительно к специфике задачи построения САПР и расчета электронных схем следует определить основные проблемы, которые потребуется разрешить [1].

1. **Работа с большим объемом ценных данных.** Большие данные (Big data) — это тренд, который еще в 2011 г. был отмечен как один из самых актуальных наравне с виртуализацией, энергосбережением и мониторингом, а с 2013 г. прочно занял свое место в индустрии.
2. **Работа с данными различных сложных типов.** В гетерогенных средах работа со сложными типами данных весьма трудоемка и требует дополнительного внимания и контроля как с серверной, так и с клиентской стороны, чтобы избежать возможных ошибок внутри исполняемого кода или ошибочного изменения либо потери данных.
3. **Передача данных, защита содержимого, транзакционные методы передачи, скорость.** В синхронной схеме «запрос–обработка–ответ» — это одна петля HTTP, которая не имеет жесткого контроля завершенности и легко может прерваться. В асинхронной — три отдельных процесса: запрос (отправлен и подтвержден), обработка (с повтором в случае сбоя), доставка ответа (с гарантией), шифрация содержимого сообщения. Асинхронная передача данных характеризуется невозможностью перехвата или подмены сообщения, прерыванием посредством дополнительных участников сети, таких как прокси-серверы, отсутствием большого количества служебной информации в запросах и ответах, не имеющей значения в бизнес-предназначении сервиса.
4. **Работа с большим числом запросов.** Создание системы, оптимизирующей фронт входящих сообщений по группе вычислительных станций и определя-

ющей загруженность сервисов путем задания последовательности действий по обеспечению параллельности вычислений и хранения запросов.

5. **Самодиагностика узлов.** Проверка целостности и качества пакета входящих данных до вхождения в узел и после его обработки при помощи системы проверки статуса самого узла
6. **Модернизация бизнес-логики.** Слабая связанность отдельных узлов (сервисов), позволяющая быстро и без потери данных заменить элемент обработки данных новым элементом так, чтобы не потребовалось полной остановки всей системы, и включающая в себя завершение выполнения начатых процессов старым кодом, для того чтобы не возникло конфликтов при обработке данных с использованием частично старого, частично нового кода.

Приведенным критериям удовлетворяет сервис-ориентированный подход (SOA) к формированию бизнес-логики продукта, успешно зарекомендовавший себя в построении информационных систем. Большой объем передаваемых сложно структурированных данных делает актуальной задачу организации взаимодействия с сервером на основе программных средств, обеспечивающих высокую скорость асинхронной передачи данных.

Цель настоящего исследования — применение сервис-ориентированной архитектуры к построению схемотехнической САПР с использованием полнодуплексного протокола связи WebSocket. Ввиду применения протокола WebSocket (стандарт RFC 6455) также требуется произвести исследование методов передачи данных по указанному протоколу.

2 Архитектура построения веб-сервиса на основе протокола WebSocket

Веб-сервис, или веб-служба, — программная система, обладающая заранее описанным интерфейсом. Для того чтобы клиенты знали, как правильно сформулировать запрос и в каком виде должны получить на него ответ, а также для взаимодействия самих сервисов с другими службами, существуют протоколы связи и упаковки данных (в классическом понимании SOAP, или REST, или HTTP без надстроек). До формирования понятия SOA (начало 2000-х гг.) построению бизнес-логики приложения не уделялось должного внимания, отсутствовало единое представление о создании веб-служб [2].

Основным отличием обобщения лучших практик создания информационных систем под названием SOA является возможность к постоянной адаптации и динамическому развитию не только путем масштабирования, но и путем собственной реорганизации и модернизации. SOA состоит из нескольких слабосвязанных компонентов, благодаря чему они могут быть легко заменены или модернизированы. Сервисная шина (enterprise service bus, ESB) служит для обмена сообщениями через единую точку, благодаря которому происходит транзакционный контроль обмена данными. SOA-реестр (SOA Registry) — электронный каталог всех



Сервис-ориентированная архитектура (SOA)

доступных на текущий момент сервисов, содержащий информацию о сервисе. К компонентам относится также модуль управления рабочими процессами (Workflow engine), содержащий модели потока выполняемых работ для определения протокола их исполнения, а также возможности параллельной и асинхронной работы над задачами. Сервисный брокер (Service broker) координирует работу бизнес-сервисов для выполнения задачи, получая информацию о сути запроса, запрашивает у Workflow engine модель работы с запросом, опрашивает сервисы через SOA-реестр (WSDL-характеристики) о готовности их к выполнению задач, запускает процесс. Одна из основных задач SOA-супервизора (SOA supervisor) — отслеживать работу различных компонентов внутри SOA-системы, оценивать корректность их функционирования, а также контролировать запросы, посланные во внешние системы. Взаимодействие всех этих компонентов показано на рисунке.

При использовании сервис-ориентированного подхода появляется возможность применять программные компоненты многократно, что позволяет снизить трудоемкость разработки схемотехнической САПР и стоимость разработки и модернизации ПО, построенного с применением архитектуры SOA.

Ранее сервис-ориентированный подход к построению веб-сервисов в основном использовался в информационных системах ввиду специфики данной области — выполнения большого числа однотипных запросов с малой нагрузкой (небольшим объемом вычислений или их отсутствием). В настоящее время SOA активно

используется и для построения САПР [3]. Данная технология уже успешно внедрена несколькими прогрессивными компаниями, занимающимися разработкой САПР, в частности решением «связанных» задач упругость–динамика–прочность, теплопередача – напряженно-деформированное состояние, моделированием мехатронных систем, вопросами разрушения композитных конструкций, сложного контактного взаимодействия, — MSC Software.

MSC SimEnterprise — семейство программных продуктов для организации и управления виртуальным моделированием, анализом и оптимизацией разрабатываемых изделий в рамках предприятия. Семейство MSC SimEnterprise включает системы MSC SimXpert, MSC SimDesigner, MSC SimManager, внедрение и применение которых обеспечивают:

- многодисциплинарный инженерный анализ в единой среде;
- накопление знаний на базе шаблонов (сценариев) решения сложных задач;
- автоматизацию выполнения инженерных расчетов (в том числе многовариантных) с отслеживанием логических связей исходных данных и получаемых результатов, а также автоматизацию генерации отчетов;
- накопление и систематизацию расчетных моделей, результатов исследований, технических отчетов и т. п.;
- контроль доступа к данным;
- организацию и контроль заданной последовательности и полноты выполнения отдельных этапов расчетных работ.

Применительно к схемотехническим САПР можно сформулировать ряд основных задач, востребованных к решению в виде предоставляемого сервиса [4]:

- расчет стационарного режима линейных схем;
- расчет частотных характеристик линейных схем;
- расчет стационарного режима нелинейных схем;
- расчет динамических характеристик нелинейных схем;
- расчет чувствительности схем к вариации параметров внешних воздействий.

Пользователю сервиса может потребоваться как решение одной из предоставляемых задач, так и различные комбинации задач по расчету схем. Например, решение задачи расчета частотных характеристик линейной схемы и расчет стационарного режима линейной схемы с последующим вычислением частотной характеристики.

Самым значительным изменением в классической модели сервис-ориентированной архитектуры на сегодняшний день является протокол передачи данных между слабосвязанными узлами. В стандартном понимании веб-службы для передачи данных используется надстройка над протоколом HTTP — протокол SOAP для обмена сообщениями в формате XML, который не лишен недостатков:

использование SOAP-протокола для передачи сообщений между узлами увеличивает их объем и снижает скорость обработки. Для увеличения скорости обычно используется пересылка XML-документов через HTTP напрямую, а сами параметры текущего запроса передаются как обычные HTTP-параметры. Хотя SOAP является стандартом, некоторые программы часто генерируют сообщения в несовместимом формате. Базовый протокол подразумевает ожидание ответа каждый раз при отправке запроса, рискуя прерваться, преодолевая двойную дистанцию от отправителя до получателя и снова до отправителя, не говоря уже о том, что запрос может быть отправлен впустую, если ответ не готов, тем самым еще раз притормозив получателя в формировании ответа своим внеочередным запросом.

Для увеличения быстродействия требуется такая организация общей шины ESB, чтобы сервисы могли отправлять и получать события асинхронно, не ожидая друг друга. Это значит, что сервис-отправитель должен иметь возможность отправлять сколько угодно событий в ESB без временных задержек и переходить к решению следующих задач. А сервисы-получатели должны получать новые события также асинхронно, независимо друг от друга, сразу же по завершении обработки предыдущего события. Решением данной проблемы может стать использование протокола двусторонней полнодуплексной связи WebSocket (стандарт RFC 6455).

У протокола SOAP и стандартного подхода к построению веб-служб есть унифицированное описание самого сервиса, несмотря на то что зачастую это описание не всегда достаточно для создания клиентского приложения или организации деятельности SOA-реестра — оно имеется и представлено в виде WSDL-файла. WSDL в зависимости от используемой версии может иметь различное оформление, но по сути несет в себе четыре основные логические части:

- (1) определение типов данных (*types*) — определение вида отправляемых и получаемых сервисом XML-сообщений;
- (2) элементы данных (*message*) — сообщения, используемые веб-сервисом;
- (3) абстрактные операции (*portType*) — список операций, которые могут быть выполнены с сообщениями;
- (4) связывание сервисов (*binding*) — способ, которым сообщение будет доставлено.

WSDL-записи могут быть созданы автоматически средой разработки путем выбора шаблона «создание веб-служб». Для проектирования сервисов на базе полнодуплексного протокола связи WebSocket требуется самостоятельно предусмотреть средства для информирования потенциальных клиентов о типах операций и протоколах, используемых данной службой в виде всем привычного WSDL-файла. Однако WSDL не позволяет разрешить все вопросы, связанные с интерфейсом работы веб-сервиса.

Проблема передачи сложных типов данных в отношении построения веб-приложения схемотехнической САПР заключается в потребности передавать большой объем сложно структурированных данных, которые в гетерогенных средах могут формироваться по разным принципам и интерпретироваться соответственно, что может послужить причиной ошибки в расчетах.

С учетом специфики протокола WebSocket, а также в целях предотвращения ошибок при интерпретации данных следует привести передаваемые данные к строковому типу String, ввести строгий порядок значений переменных и ограничить его рамками — специальными зарезервированными символами, благодаря которым определяется, к какому объекту относится данное информационное описание. Таким образом, на сервер передается не многомерная структура, а строка, где каждый коэффициент имеет свое строгое место, а данные, относящиеся к разным объектам, разделяются специальными символами. Например, строка « $1*2*0,1*2*3*1*2*0*10*3*0*1*1*0*3*0*0,16*0*0*1*$ » заключает в себе данные о резисторах, катушке индуктивности, емкостном элементе, входных и выходных узлах, а также параметрах расчета.

Для того чтобы расшифровать строку, необходимо иметь специальный ключ, который в своем прикладном варианте представляет класс или функцию Parser (интерпретатор). Очевидно, что сервер имеет такую функцию или класс для декодирования данных, для разработчика клиентской составляющей данный ключ неизвестен, его следует разместить в WSDL-файле. Альтернативным вариантом предоставления информации о сервисе может послужить описание методологии работы веб-службы на сайте.

Для построения веб-сервиса на основе сервис-ориентированного подхода с применением полнодуплексного протокола WebSocket нужно учитывать две составляющие: построение шины ESB на основе асинхронного протокола и построение правил и протокола взаимодействия с внешней средой также на основе данного типа связи.

В первом случае требуется WSDL-файл для SOA-реестра, который можно организовать через специальную структуру (похожую по информационному наполнению на WSDL), подходящую для данного конкретного случая, и разместить данный компонент в виде модели в узле Workflow engine для возможности быстро заменить его. При этом при замене сервиса на новый можно предусмотреть в структуре информации о сервисе обратную совместимость для старого компонента и нового соответственно [5].

Во втором случае, когда протокол связи WebSocket используется от точки входа до конечного пользователя, в идеальной ситуации потребуется предусмотреть вариант запроса WSDL-файла через URL и организовать возможность предоставления структуры WSDL-файла на обозрение конечному пользователю.

При рассмотрении протокола WebSocket для построения ESB-шины веб-сервиса видится ряд преимуществ, связанных в первую очередь со скоростью обмена данными [6]:

- возможность асинхронного обмена данными между сервисами;
- возможность проведения асинхронной работы над исполняемыми задачами;
- отсутствие потребности в постоянном опросе сервисов по HTTP/SOAP-петле (уменьшение передаваемого трафика за счет снижения числа запросов, уменьшение нагрузки на сам опрашиваемый сервис);
- уменьшение трафика по сети за счет снижения объема самих пакетов (особенность протокола WebSocket);
- активация соединения требуется единственный раз при активации веб-сервиса либо после внесения изменений в компоненты сервиса;
- дублирование точки входа, поскольку WebSocket можно дублировать неограниченное число раз, так как общение между ним и клиентом унифицировано и отвязано от внутренностей системы, а общение между ним и системой, в свою очередь, отвязано от коммуникаций с клиентом.

С точки зрения использования протокола WebSocket для обмена информацией с клиентским приложением также имеется ряд преимуществ:

- интерактивность приложения за счет асинхронного подхода к обмену пакетами;
- отсутствие лишней нагрузки на сервер за счет посылаемых запросов (ответ придет сам тогда, когда будет готов);
- уменьшение трафика по сети за счет уменьшения объема самих пакетов (особенность протокола WebSocket).

Подобный подход к построению веб-служб соответствует всем шести сформулированным выше критериям, предъявляемым к современным глобальным веб-сервисам — отказоустойчивым, удобным и обладающим высоким быстродействием.

3 Заключение

Использование полнодуплексного протокола связи WebSocket для построения крупных сервисов на основе сервис-ориентированной архитектуры дает много преимуществ по сравнению со стандартным методом организации веб-служб (протокол SOAP). WebSocket — это актуальное и логичное дополнение к классической модели SOA, позволяющее расширить ее возможности: ускорить обмен данными, ускорить решение задач за счет асинхронности выполнения процесса разными компонентами; кроме того, увеличить надежность работы сервиса за счет уменьшения числа запросов для получения статуса и информации (ответ будет тогда, когда будет готов). При этом расширить возможности масштабирования за счет возможности дублирования точки входа.

Существуют и отрицательные стороны применения данного протокола — ручное формирование информационной поддержки о сервисе в том или ином

виде. По мере распространения и возможной модернизации протокола такой недостаток, возможно, будет исправлен.

Литература

1. *Anisimov V. I., Almaasali S. A. Methods of organization of computations during the simulation of grand systems based on diacoptics // IEEE Proceedings. North West Section, 2014. Vol. 6. P. 44–48.*
2. *Гридин В. Н., Дмитревич Г. Д., Анисимов Д. А. Архитектура распределенных сервис-ориентированных систем автоматизированного проектирования // Известия Южного федерального университета. Технические науки, 2014. № 7(156). С. 51–58.*
3. *Гридин В. Н., Анисимов В. И. Методы построения распределенных систем автоматизированного проектирования // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM-2015): Мат-лы XV Междунар. конф. — М.: Аналитик, 2015. С. 250–254.*
4. *Гридин В. Н., Анисимов В. И. Моделирование больших систем на основе методов декомпозиции и компактной обработки разреженных матриц // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM-2015): Мат-лы XV Междунар. конф. — М.: Аналитик, 2015. С. 62–66.*
5. *Анисимов В. И., Гридин В. Н., Васильев С. А. Построение веб-приложений на основе полнодуплексного протокола передачи данных WebSocket в сервис-ориентированных системах автоматизации схемотехнического проектирования // Информационные технологии и математическое моделирование систем 2015: Тр. Междунар. науч.-технич. конф. — М.: Центр информационных технологий в проектировании РАН, 2015. С. 26–28.*
6. *Федоренков Р. В., Ничушкина Т. Н. Интерактивный веб-сервис WebSocket 2307-0595 // Инженерный вестник, 2015. № 01. С. 539–545.*

Поступила в редакцию 22.01.16

METHODS OF CREATION OF A CIRCUITRY CAD USING THE SERVICE-ORIENTED APPROACH ON THE BASIS OF THE WEBSOCKET PROTOCOL

V. I. Anisimov^{1,2}, V. N. Gridin², and S. A. Vasilev¹

¹Saint Petersburg Electrotechnical University “LETI,” 5 Professora Popova Str., St. Petersburg 197376, Russian Federation

²Design Information Technologies Center, Russian Academy of Sciences, 7a Marshal Biryuzov Str., Odintsovo, Moscow Region 143000, Russian Federation

Abstract: The main criteria to which the web service of circuitry design shall correspond are given. The classical approach to creation of service-oriented systems is described. The article reveals the internal architecture of a web

service on the basis of service-oriented architecture (SOA) and the main stages of functioning. The article describes application of SOA for automated design engineering systems, in particular circuitry CAD (computer-aided design). An example of successful implementation of the SOA architecture in automated design engineering systems is given. A number of circuitry tasks for creation of an automated design engineering system on the basis of the service-oriented approach are formulated and described. The article considers the advantages and shortcomings of use of the WebSocket full-duplex communication protocol (RFC 6455 standard) as the basis of an ESB bus of a service-oriented automated design engineering system in comparison with the SOA protocol. The advantages and shortcomings of use of the WebSocket protocol for performing external communications of a web service are given. The question of transmission of complex data types in heterogeneous software environments using the WebSocket full-duplex data protocol (RFC 6455 standard) is taken up. The advantages and shortcomings of implementation of the WebSocket communication protocol in automated design engineering systems, in particular circuitry design systems, on the basis of SOA are specified.

Keywords: SOA; CAD; web service; WebSocket; environment .NET; bus ESB; cross-platform; asynchrony; heterogeneous environment; frequency response; stationary mode; circuit engineering; difficult data types; network traffic; high-speed performance

DOI: 10.14357/08696527160209

Acknowledgments

The research was financially supported by the Program of Basic Research of the RAS Department for Nanotechnologies and Information Technologies (ONIT) "Scientific basis for the creation of heterogeneous telecommunication and location systems and their elemental base."

References

1. Anisimov, V. I., and S. A. Almaasali. 2014. Methods of organization of computations during the simulation of grand systems based on diacoptics. *IEEE Proceedings, North West Section*. 6:44–48.
2. Gridin, V. N., G. D. Dmitrevich, and D. A. Anisimov. 2014. Arkhitektura raspredelennykh servis-orientirovannykh sistem avtomatizirovannogo proektirovaniya [Architecture distributed service-oriented CAD]. *Izvestiya Yuzhnogo Federal'nogo Universiteta. Tekhnicheskie Nauki* [Bulletin of the Southern Federal University. Technical Science] 7(156):51–58.
3. Gridin, V. N., and V. I. Anisimov. 2014. Metody postroeniya raspredelennykh sistem avtomatizirovannogo proektirovaniya [Methods of creation of distributed systems of CAD]. *15th Conference (International) "Systems of Design, Technological Preparation of Production and Control of Stages of Life Cycle of an Industrial Product (CAD/CAM/PDM)" Proceedings*. Moscow: Analitik. 250–254.

4. Gridin, V. N., and V. I. Anisimov. 2015. Modelirovanie bol'shikh sistem na osnove metodov dekompozitsii i kompaktnoy obrabotki razrezhennykh matriks [Simulation of big systems on the basis of decomposition techniques and compact processing of disperse matrixes]. *15th Conference (International) "Systems of Design, Technological Preparation of Production and Control of Stages of Life Cycle of an Industrial Product (CAD/CAM/PDM)" Proceedings*. Moscow: Analitik. 62–66.
5. Anisimov, V. I., V. N. Gridin, and S. A. Vasilev. 2015. Postroenie veb-prilozheniy na osnove polnodupleksnogo protokola peredachi dannykh WebSocket v servis-orientirovannykh sistemakh avtomatizatsii skhemotekhnicheskogo proektirovaniya [Creation of web applications on the basis of the full-duplex data transfer protocol WebSocket in service-oriented systems of automation of circuitry design]. *Scientific and Technical Conference (International) on Information Technologies and Mathematical Simulation of Systems Proceedings*. Moscow: Tsentr Informatsionnykh Tekhnologiy v Proektirovaniy RAN. 26–28.
6. Fedorenkov, R. V., and T. N. Nichushkina. 2015. Interaktivnyy veb-servis WebSocket 2307-0595 [Interactive web service of WebSocket 2307-0595]. *Inzhenernyy Vestnik* [Engineering Bulletin] 1:539–545.

Received January 22, 2016

Contributors

Anisimov Vladimir I. (b. 1926) — Doctor of Science in technology, professor, Saint Petersburg Electrotechnical University “LETI,” 5 Professora Popova Str., St. Petersburg 197376, Russian Federation; principal scientist, Design Information Technologies Center, Russian Academy of Sciences, 7a Marshal Biryuzov Str., Odintsovo 143000, Moscow Region, Russian Federation; vianisimov@inbox.ru

Gridin Vladimir N. (b. 1944) — Doctor of Science in technology, professor, scientific leader, Design Information Technologies Center, Russian Academy of Sciences, 7a Marshal Biryuzov Str., Odintsovo, Moscow Region 143000, Russian Federation; info@ditc.ras.ru

Vasilev Sergey A. (b. 1991) — PhD student, Saint Petersburg Electrotechnical University“LETI,” 5 Professora Popova Str., St. Petersburg 197376, Russian Federation; venom-gt@list.ru

НОРМАЛИЗАЦИЯ БАНКОВСКИХ ОНЛАЙН-СЕРВИСОВ

В. Д. Ильин¹

Аннотация: Представлены основные задачи, решаемые с помощью банковских онлайн-сервисов нормализованного экономического механизма (НЭМ). Приведен состав основных онлайн-сервисов банка-регулятора, банков-прайдеров и персональных электронных банков. Более подробно рассмотрены онлайн-сервисы государственного удостоверения операций над банковскими счетами, досрочного возврата долгов и долговой денежной эмиссии.

Ключевые слова: нормализованный экономический механизм; банк-регулятор; банк-прайдер; персональный электронный банк; банковский онлайн-сервис

DOI: 10.14357/08696527160210

1 Введение

Результативность функционирования современного *экономического механизма* (ЭМ) во многом определяется качеством банковских онлайн-сервисов, информационного обслуживания *участников экономической деятельности* (эд-участников) и документального отображения сделок и имущественных статусов эд-участников. Изменения в экономических отношениях и информационных технологиях должны отражаться в изменениях устройства ЭМ, включая банковские онлайн-сервисы.

Вряд ли экономисты, поддерживающие действующую *экономическую модель экстерриториальной выгоды* с банковской системой, способствующей финансово-му мошенничеству, согласятся признать справедливость этого утверждения. Привыкшие рассуждать о хозяйственной деятельности как о слабоизученной стихии («кризис разразился» и т. п.) и азартной игре («игра на повышение» и пр.), они продолжают отстаивать экономические модели [1–6], адекватность которых невозможно оценить, как это принято в науке и технике.

Развернутый критический анализ экономической модели экстерриториальной выгоды и описание альтернативной модели *нормализованной экономики* [7–12], ориентированной на развитие национального потенциала, приведены в [13].

В наши дни российские банки продолжают развивать онлайн-сервисы «по своему усмотрению», а Банк России уклоняется от руководящей роли в выработке требований к составу и качеству функционирования банковских онлайн-сервисов.

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vdilyin@ipiran.ru

Унификация технологии дистанционного банковского обслуживания на основе онлайн-сервисов предполагает поэтапную унификацию банковской инфраструктуры (ее властно-координационной составляющей, онлайн-сервисов, банковских сетей и др.), которая в наши дни не менее важна, чем транспортная инфраструктура.

А пока в российской банковской практике продолжается способствующее банковскому мошенничеству выполнение операций над банковскими счетами без обязательных онлайн-подтверждений допустимости операций, предварительно полученных от владельцев счетов. Банки уклоняются от предоставления онлайн-сервиса платежей по схеме: \langle платежное поручение плательщика с указанием реквизитов его счета, счета получателя платежа и суммы платежа $\rangle \Rightarrow \langle$ подтверждение плательщика (в ответ на запрос сервиса подтвердить платежное поручение) $\rangle \Rightarrow \langle$ подтверждение получателя платежа о согласии получить платеж $\rangle \Rightarrow \langle$ незамедлительное перечисление указанной суммы со счета плательщика на счет получателя платежа $\rangle \Rightarrow \langle$ онлайн-квитанция плательщику о выполнении платежа \rangle .

1.1 Выделение фрагментов текста статьи

Для выделения определений, замечаний и примеров используются следующие средства:

- \square \langle фрагмент описания \rangle $\square \approx$ утверждение (определение, аксиома и др.) (здесь и далее символ \approx заменяет слово «означает»);
- \diamond \langle фрагмент описания \rangle $\diamond \approx$ замечание;
- \circlearrowleft \langle фрагмент описания \rangle $\circlearrowleft \approx$ пример.

Курсивом выделены первые вхождения названий понятий и фрагменты описания, к которым автор хочет привлечь внимание.

1.2 Научно-исследовательская работа, результаты которой представлены в статье

Результаты, представленные в статье, получены в процессе выполнения научно-исследовательской работы (НИР) «Создание методологии информатизации нормализованного экономического механизма и программная реализация комплекса экспертизного планирования ресурсов на основе интернет-сервисов».

Первая очередь средств информатизации *нормализованного экономического механизма* включает комплекс интернет-сервисов экспертизного планирования ресурсов. В 2015 г. завершены работы по созданию теоретических основ методологии информатизации НЭМ [13]. Опубликованы усовершенствованный метод решения задачи, принципы реализации и описание применения *интернет-сервиса «Планирование расходов»* [14]. С методологическим основанием и применением этого сервиса связаны результаты, опубликованные в [15–18].

2 Онлайн-сервис: определение и истолкование

□ **Онлайн-сервис** (англ. online service) — комплекс онлайн-услуг определенного назначения (электронная почта, поиск и др.), реализованный в *s-среде* [19]. □

Онлайн-сервисы — воплощение информатизации [19] «под ключ» различных видов деятельности [образовательной (дистанционное образование и пр.), экономической (банковские сервисы и пр.) и др.]. Пользователь онлайн-сервиса, заплатив сервис-провайдеру, получает выбранные им услуги (в виде решения интересующих его задач), освобождаясь от необходимости разбираться в аппаратных и программных средствах *s-машин* [19], заботиться об апгрейдах и апдейтах.

Интернет-сервисы — часть онлайн-сервисов *s-среды* (○ Веб, электронная почта, IP-телефония, поисковики, сервисы социальных сетей и др. ○). Другая часть онлайн-сервисов (не использующая Интернет) функционирует в сетях государственных ведомств, корпораций и индивидуальных предпринимателей [19].

3 Нормализованная банковская система и ее онлайн-сервисы

Банковская система нормализованного экономического механизма включает *персональные электронные банки физических лиц* (ПЭБы), *корпоративные электронные банки* (КЭБы), *банки-провайдеры* и *банк-регулятор*, под управлением которого функционируют все другие банки [13].

Ядром нормализованной банковской системы является *система имущественных статусов* (ис-система) — программно реализуемая в *s-среде* система документального представления денежной и неденежной составляющих имущественных статусов *участников экономической деятельности*. Денежная составляющая выражена значениями сумм нормализованных денег в разделах *的独特ых глобальных счетов* эд-участников (ис-счетов).

Неденежная составляющая имущественного статуса эд-участника представлена документами, подтверждающими право собственности на недвижимость, транспорт и другое имущество, которое при необходимости может рассматриваться как залоговое.

Нормализованные деньги (НД) — универсальное электронное средство количественного неанонимного документирования имущественных отношений, удостоверяемых государством.

Нормализованные деньги — не альтернативная валюта, а *объединение национальных валют, правила эмиссии и применения которых нормализованы*. Нормализованные деньги предназначены для представления стоимости товаров и имущественных статусов эд-участников, для оплаты товаров, договорного денежного инвестирования и накопления богатства; НД-суммы представлены записями в ис-счетах, удостоверяющими имущественное право на долю стоимости товаров экономической системы и имущественные обязательства по отношению к другим эд-участникам, выраженные суммами долгов и полученных инвестиций.

Универсальность НД выражается в применимости во всех типах внутренних и внешних экономических сделок, разрешенных законом государства, под юрисдикцией которого функционирует экономическая система.

Неанонимность НД состоит в том, что не существует НД-сумм в ис-счетах, принадлежность и происхождение которых не были бы документально определены. Каждая из них отражает результат *допустимой сделки* (торговой, инвестиционной или др.) между конкретными эд-участниками [13].

В каждый момент времени совокупная НД-сумма, записанная в ис-счетах, обеспечена реальными товарами, предлагаемыми на поливалютном рынке, где рыночная стоимость товаров является итогом стоимостных компромиссов продавцов с покупателями и определяется соотношением спроса и предложения.

Для НД-сумм определены два состояния: «*назначенные*» (долг, образовавшийся при покупке товара; инвестиция; налог или другая сумма, которую необходимо заплатить) и «*неназначенные*» (суммы в разделах «Располагаю» счетов). «Назначенные» НД-суммы могут использоваться только по определенному назначению [○ полученные от инвесторов — в соответствии с инвестиционными договорами (закупка нового оборудования и т. д.) ○]. «Неназначенные» НД-суммы используются по усмотрению владельца счета (в любых допустимых сделках).

Ис-счет — уникальный поливалютный банковский счет эд-участника (физического или юридического лица). Представляет собой унифицированный электронный документ, состоящий из *валютных частей* (активируемых банком-регулятором). Каждая из них имеет разделы: «Располагаю», «Кредитование», «Инвестирование», «Налоги и сборы», «Дарение», «Пожертвование». Совокупность допустимых операций (при документировании сделок) определяется принадлежностью суммы к подразделу счета (○ сумма из подраздела «Получено» раздела «Инвестирование» может использоваться только для оплаты товаров, типы которых перечислены в инвестиционных договорах ○). Поскольку ис-счет имеет *поливалютную структуру*, он применим для документирования результатов внутренней и внешней экономической деятельности. Существование ис-счета предполагает, что его владелец имеет *的独特的全球性身份证件* [13].

3.1 Банк-регулятор и его онлайн-сервисы

□ *Банк-регулятор* — государственное учреждение, управляющее функционированием банковской системы НЭМ.

Задачи банка-регулятора, реализуемые посредством онлайн-сервисов:

- выдача и отзыв лицензий на право банковской деятельности (владельцам ПЭБов, КЭБов и банков-провайдеров);
- активация и деактивация валютных частей ис-счетов;
- государственное удостоверение операций над ис-счетами эд-участников;

- обслуживание запросов на досрочный возврат долгов и выполнение долговой денежной эмиссии (при реализации технологии долговой торговли);
- контроль выполнения эд-участниками правил банковской деятельности;
- анализ финансовой составляющей экономической деятельности и представление результатов в порядке, установленном законом;
- разработка, модификация и утверждение прошедших тестирование унифицированных форм банковских документов (включая ис-счета);
- контроль эффективности размещения средств *государственного денежного резерва, фондов государственной социальной защиты*, других государственных фондов и фондов с государственным участием. □

Банк-регулятор располагает сетью серверов, размещенных на территории страны, под юрисдикцией которой находится экономическая система.

3.1.1 *Онлайн-сервис государственного удостоверения операций над ис-счетами*

Изменение состояния ис-счета может быть реализовано только после получения *государственного удостоверения допустимости операции* над ис-счетом. Удостоверение выдает специальный онлайн-сервис банка-регулятора. Основанием для получения удостоверения служат данные совершающей операции (сделка купли-продажи, договорного инвестирования или др.).

Таким способом исключаются возможности нарушения установленных законом правил товарно-денежного обращения (недопустимые экономические сделки, неуплата установленных законом государственных сборов и др.).

3.1.2 *Онлайн-сервис досрочного возврата долгов и долговой денежной эмиссии*

В общем случае возврат долга покупателя продавцу определяется их документально оформленным договором. Применительно к долгам, образовавшимся в результате продажи приоритетных жизненно-необходимых товаров, могут действовать правила досрочного возврата долгов продавцам из средств банка-регулятора. Эти правила определяются банком-регулятором и действуют применительно к продавцам товаров, входящих в утвержденный законом приоритетный список. Продавцам таких товаров долговая часть стоимости покупки может быть возвращена в относительно короткие сроки (раньше, чем предусмотрено договорным графиком возврата долга). Делается это по запросам продавцов в *долговой отдел банка-регулятора*, направляемых с использованием соответствующего онлайн-сервиса. При положительном решении банк-регулятор переводит на ис-счет продавца сумму долга, а покупатель после этого выплачивает долг не продавцу, а долговому отделу банка-регулятора.

Долговая денежная эмиссия выполняется банком-регулятором только тогда, когда сумма возвращенных покупателями долгов меньше очередной долговой

суммы, запрошенной для возврата продавцу. Эмитируемая сумма равна разности запрошенной долговой суммы и суммы на счету долгового отдела.

Долговая денежная эмиссия служит средством *государственного регулирования общей суммы нормализованных денег в экономической системе*. Таким способом осуществляется государственное регулирование товарной емкости денег.

◊ Правила эмиссии долговых сумм исключают возможность эмиссии денежных сумм, не обеспеченных товарами. ◊

3.2 Банки-провайдеры и их онлайн-сервисы

□ *Банки-провайдеры* — коммерческие предприятия, учрежденные юридическими лицами (или объединениями юридических лиц, объединениями физических лиц, объединениями юридических и физических лиц), занимающимися производством товаров или их реализацией.

Онлайн-сервисы банка-провайдера предназначены:

- для обслуживания запросов владельцев ис-счетов, направляемых посредством ПЭБов и КЭБов при совершении сделок (включая запросы на удостоверение состояния ис-счетов партнеров по сделкам, направляемые с их разрешения);
- хранения и обновления зашифрованных копий ис-счетов клиентов банка-провайдера;
- анализа инвестиционных запросов клиентов (потенциальных инвесторов и получателей инвестиций); банки-провайдеры могут выполнять заказы получателей инвестиций на консолидацию заявок инвесторов, чтобы комплексировать заказанную сумму инвестиции;
- регистрации заключенных договоров (с контролем допустимости сделок) и ведения баз данных таких договоров;
- юридического сопровождения реализуемых сделок. □

В базах данных банков-провайдеров в зашифрованной форме хранятся копии документов по всем обслуженным сделкам (за период времени, установленный законом).

Банк-провайдер располагает объединенными в сеть серверами, предназначенными для обслуживания запросов от ПЭБов и КЭБов клиентов и взаимодействия с серверами банка-регулятора. Для устойчивого функционирования НЭМ целесообразно, чтобы учредителями банков-провайдеров были юридические и физические лица, занимающиеся производством, распределением и резервированием реальных товаров и имеющие для этого достаточные не только денежные, но и неденежные имущественные основания. Правила, определяющие процессы учреждения, функционирования и ликвидации банков-провайдеров, должны быть установлены законом.

3.3 Персональные и корпоративные электронные банки

Персональные и корпоративные электронные банки (как программно-аппаратные средства) — это, как правило, переносные с-машины (типа планшетов), наделенные функциями смартфонов. Корпоративные электронные банки могут быть реализованы и на основе стационарных с-машин. В ПЭБах и КЭБах в зашифрованной форме хранятся оригиналы ис-счетов и документов по сделкам. Комплекс программ мобильного банкинга — ядро приложений ПЭБа или КЭБа. Зашифрованная база данных ис-счета хранится в памяти устройства, а ее копия — на карте памяти. Программно реализуемые записи в файлах ис-счетов имеют право инициировать только владельцы ис-счетов, совершающие сделку. Запросы на обслуживание сделки (удостоверение состояния ис-счета и сохранение обновленной копии ис-счета) получают банки-провайдеры, сервисами которых пользуются совершающие сделку эд-участники.

4 Онлайн-сервис подтверждения операций над ис-счетами

Этот сервис реализует проверку выполнения аксиомы допустимости операций над ис-счетами.

□ Никакое изменение сумм ис-счета не может быть осуществлено без *документируемого зашифрованного подтверждения владельца ис-счета* (а в неординарных ситуациях — после документируемого зашифрованного подтверждения определенного законом государственного органа). □

В НЭМ этот сервис предоставляется всеми банками (ПЭБами, КЭБами, банками-провайдерами и банком-регулятором) и реализуется на программно-аппаратном уровне. Операция над ис-счетом становится допустимой (при сделках купли-продажи, инвестирования и др.) только тогда, когда от владельца каждого ис-счета, вовлеченного в операцию, получено зашифрованное согласие. Каждое такое подтверждение входит в состав документа, отражающего операцию над ис-счетом. Документ получают владельцы ис-счетов; его копия хранится установленное время на серверах банков-провайдеров [и при совершении ряда операций (○ в частности, досрочного возврата долгов ○) — на серверах банка-регулятора].

5 О прогнозе изменений в банковской системе

◊ Прошло более пяти лет со времени русскоязычной публикации первой версии *модели нормализованной экономики* [7] и более двух лет со времени англоязычной публикации второй версии этой модели [9], в каждой из которых было приведено и описание *персональных электронных банков*.

В начале 2015 г. Билл Гейтс с супругой сообщили об их прогнозе изменений в банковской системе к 2030 г. «В течение 15 лет цифровой банкинг даст бедным больший контроль над их имуществом и поможет им изменить жизнь. Ключом к этому станут мобильные телефоны. К 2030 г. два миллиарда человек, у которых

сейчас нет банковского счета, смогут использовать телефон для управления счетом и отправки платежей. А затем мобильные провайдеры обеспечат весь спектр услуг, включая депозиты, кредиты и страховку» [20]. ◇

6 Заключение

Банк-регулятор и банки-провайдеры *нормализованной экономической системы* служат ядром системы документирования результатов экономической деятельности. При совершении сделок их онлайн-сервисы обрабатывают запросы эд-участников, выполняемые с помощью КЭБов и ПЭБов. Онлайн-сервис подтверждения операций над ис-счетами предоставляется всеми банками НЭМ.

Для эффективной работы НЭМ особое значение имеют следующие онлайн-сервисы банка регулятора:

- *государственное удостоверение операций над ис-счетами эд-участников;*
- обслуживание запросов на *досрочный возврат долгов и выполнение долговой денежной эмиссии* (при реализации технологии долговой торговли [12, 13]).

Литература

1. Fisher I. The purchasing power of money. Its determination and relation to credit, interest, and crises. — New York, NY, USA: Macmillan Co., 1922. 548 p.
2. Keynes J. M. A tract on monetary reform. — London: Macmillan Co., 1923. 220 p.
3. Keynes J. M. The general theory of employment, interest and money. — Macmillan Cambridge University Press, for Royal Economic Society, 1936. 416 p. <http://www.marxists.org/reference/subject/economics/keynes/general-theory>.
4. Friedman M. A theoretical framework for monetary analysis // J. Polit. Econ., 1970. Vol. 78. No. 2. P. 193–238.
5. Friedman M. Nobel lecture: Inflation and unemployment // J. Polit. Econ., 1977. Vol. 85. No. 3. P. 451–472.
6. Friedman M. The optimum quantity of money. — Rev. ed. — AldineTransaction, 2005. 308 p.
7. Ильин В. Д. Модель нормализованной экономики (НЭк-модель): основы концепции // Управление большими системами, 2009. Вып. 25. С. 116–138.
8. Ильин А. В., Ильин В. Д. S-экономика: механизм хозяйствования в эпоху Интернета. — М.: ИПИ РАН, 2011. 105 с. https://www.researchgate.net/publication/259654393_S-ekonomika_mehanizm_hozajstvovaniya_v_epohu_Interneta_S-economics_economy_in_the_Internet_era.
9. Ilyin V. D. S-economics. — М.: ИПИ РАН, 2012. 54 с. https://www.researchgate.net/publication/258218716_S-economics.
10. Ilyin A. V., Ilyin V. D. Informatization of normalized economic mechanism: E-services for resource planning // Системы и средства информатики, 2014. Т. 24. № 4. С. 246–260.

11. *Ilyin A. V., Ilyin V. D.* Towards a normalized economic mechanism based on E-services // Agris on-line Papers in Economics and Informatics, 2014. Vol. 6. No. 3. P. 39–49. http://online.agris.cz/files/2014/agris_on-line_2014_3_ilyin_ilyin.pdf.
12. *Ilyin A. V., Ilyin V. D.* E-trade with direct lending and normalized money // Agris on-line Papers in Economics and Informatics, 2015. Vol. 7. No. 4. P. 57–64. http://online.agris.cz/files/2015/agris_on-line_2015_4_ilyin_ilyin.pdf.
13. Ильин А. В., Ильин В. Д. Информатизация экономического механизма. — М.: ИПИ РАН, 2015. 130 с. https://www.researchgate.net/publication/292059684_Informatizacia_ekonomiceskogo_mehanizma_Informatization_of_economic_mechanism.
14. Ильин А. В. Интернет-сервис планирования расходов // Системы и средства информатики, 2015. Т. 25. № 2. С. 111–122.
15. Ильин А. В. Экспертное планирование ресурсов. — М.: ИПИ РАН, 2013. 58 с. https://www.researchgate.net/publication/259654243_Ekspertnoe_planirovanie_resursov_Expert_Resource_Planning.
16. *Ilyin A. V., Ilyin V. D.* The technology of interactive resource allocation in accordance with the customizable system of rules // Appl. Math. Sci., 2013. Vol. 7. No. 143. P. 7105–7111. doi: 10.12988/ams.2013.311649.
17. *Ilyin A. V., Ilyin V. D.* The interval method of cost planning and its implementation in the online service // Contemporary Eng. Sci., 2014. Vol. 7. No. 20. P. 931–938. doi: 10.12988/ces.2014.48114.
18. *Ilyin A. V., Ilyin V. D.* Interval planning the supplies of scarce product // Contemporary Eng. Sci., 2015. Vol. 8. No. 31. P. 1495–1498. doi: 10.12988/ces.2015.59263.
19. Ильин А. В., Ильин В. Д. Основы теории s-моделирования. — М.: ИПИ РАН, 2009. 143 с. <http://smodeling.files.wordpress.com/2010/01/basics-theory-sm-20092.pdf>.
20. Ильин В. Д. Персональные банки: и Билл Гейтс не сомневается. — CNews Клуб, 2015. http://club.cnews.ru/blogs/entry/personalnye_banki_i_bill_gejts_ne_somnevaetsya.

Поступила в редакцию 14.03.16

NORMALIZATION OF ONLINE BANKING SERVICES

V. D. Ilyin

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: The paper presents basic problems to be solved by means of online banking services of the normalized economic mechanism. Primary online services of the central bank, banks-providers, and personal electronic banks are enumerated. More details are given for the online services of state certification of operations on bank accounts, early repayment of debts, and emission of debt amounts.

Keywords: normalized economic mechanism; central bank; bank-provider; personal electronic bank; online banking service

DOI: 10.14357/08696527160210

References

1. Fisher, I. 1922. *The purchasing power of money. Its determination and relation to credit, interest, and crises.* — New York, NY: The Macmillan and Co. 548 p.
2. Keynes, J. M. 1923. *A tract on monetary reform.* — London: Macmillan and Co. 220 p.
3. Keynes, J. M. 1936. *The general theory of employment, interest and money.* Macmillan Cambridge University Press, for Royal Economic Society. 416 p. Available at: <http://www.marxists.org/reference/subject/economics/keynes/general-theory> (accessed March 14, 2016).
4. Friedman, M. 1970. A theoretical framework for monetary analysis. *J. Polit. Econ.* 78(2):193–238.
5. Friedman, M. 1977. Nobel lecture: Inflation and unemployment. *J. Polit. Econ.* 85(3):451–472.
6. Friedman, M. 2005. *The optimum quantity of money.* Rev. ed. AldineTransaction. 308 p.
7. Ilyin, V. D. 2009. Model' normalizovannoy ekonomiki (NEk-model'): Osnovy kontseptsii [The model of normalized economics (NEc-model): Basics of framework]. *Upravlenie Bol'shimi Sistemami* [Large-Scale Systems Control (Russia)] 25:116–138. Available at: <http://www.mathnet.ru/links/c6cd01830a66b4c9d14eb6ce4365e59e/ubs6.pdf> (accessed March 14, 2016).
8. Ilyin, A. V., and V. D. Ilyin. 2011. *S-ekonomika: Mekhanizm khozyaystvovaniya v epokhu Interneta* [S-economics: Mechanism of management in the Internet epoch]. — Moscow: Institute of Informatics Problems of RAS. 105 p. Available at: https://www.researchgate.net/publication/259654393_S-ekonomika_mehanizm_hozajstvovaniya_v_epohu_Interneta_S-economics_economy_in_the_Internet_era (accessed March 14, 2016).
9. Ilyin, V. D. 2012. *S-economics.* Moscow: Institute of Informatics Problems of RAS. 54 p. Available at: https://www.researchgate.net/publication/258218716_S-economics (accessed March 14, 2016).
10. Ilyin, A. V., and V. D. Ilyin. 2014. Informatization of normalized economic mechanism: E-services for resource planning. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 24(4):246–260. Available at: <http://www.mathnet.ru/links/9b17f1a5fc7ff02141a015e3ab07a98e/ssi386.pdf> (accessed March 14, 2016).
11. Ilyin, A. V., and V. D. Ilyin. 2014. Towards a normalized economic mechanism based on E-services. *Agris on-line Papers in Economics and Informatics* 6(3):39–49. Available at: http://online.agris.cz/files/2014/agris_on-line_2014_3_ilyin_ilyin.pdf (accessed March 14, 2016).
12. Ilyin, A. V., and V. D. Ilyin. 2015. E-trade with direct lending and normalized money. *Agris on-line Papers in Economics and Informatics* 7(4):57–64. Available at: http://online.agris.cz/files/2015/agris_on-line_2015_4_ilyin_ilyin.pdf (accessed March 14, 2016).
13. Ilyin, A. V., and V. D. Ilyin. 2015. *Informatizatsiya ekonomiceskogo mekhanizma* [Informatization of economic mechanism]. Moscow: Institute of Informatics Problems of RAS. 130 p. Available at: <https://www.researchgate.net/publication/>

- 292059684_Informatizacia_ekonomiceskogo_mehanizma_Informatization_of_economic_mechanism (accessed March 14, 2016).
14. Ilyin, A. V. 2015. Internet-servis planirovaniya raskhodov [The online service for cost planning]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 25(2):111–122. Available at: <http://www.mathnet.ru/links/dc50949b18b78eebbb31eeb620a2f096/ssi410.pdf> (accessed March 14, 2016).
 15. Ilyin, A. V. 2013. *Ekspernoe planirovanie resursov* [Expert resource planning]. Moscow: Institute of Informatics Problems of RAS. 58 p. Available at: https://www.researchgate.net/publication/259654243_Ekspernoe_planirovanie_resursov_Expert_Resource_Planning (accessed March 14, 2016).
 16. Ilyin, A. V., and V. D. Ilyin. 2013. The technology of interactive resource allocation in accordance with the customizable system of rules. *Appl. Math. Sci.* 7(143):7105–7111. doi: 10.12988/ams.2013.311649.
 17. Ilyin, A. V., and V. D. Ilyin. 2014. The interval method of cost planning and its implementation in the online service. *Contemporary Eng. Sci.* 7(20):931–938. doi: 10.12988/ces.2014.48114.
 18. Ilyin, A. V., and V. D. Ilyin. 2015. Interval planning the supplies of scarce product. *Contemporary Eng. Sci.* 8(31):1495–1498. doi: 10.12988/ces.2015.59263.
 19. Ilyin, A. V., and V. D. Ilyin. 2009. *Osnovy teorii s-modelirovaniya* [Basics of the theory of s-modeling]. Moscow: Institute of Informatics Problems of RAS. 143 p. Available at: https://www.researchgate.net/publication/259654624_Osnovy_teorii_s-modelirovaniya_Basics_of_the_Theory_of_S-modeling (accessed March 14, 2016).
 20. Ilyin, V. D. 2015. *Personal'nye banki: I Bill Geits ne somnevaetsya* [Personal banks: And Bill Gates has no doubt]. CNews Club. Available at: http://club.cnews.ru/blogs/entry/personalnye_banki_i_bill_geits_ne_somnevaetsya (accessed March 14, 2016).

Received March 14, 2016

Contributor

Ilyin Vladimir D. (b. 1937) — Doctor of Science in technology, Head of Laboratory, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russia; vdilyin@ipiran.ru

КОГНИТИВНЫЕ АСПЕКТЫ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Г. В. Лукьянов¹, Д. А. Никишин²

Аннотация: Статья посвящена проблематике моделирования сложных информационных систем (ИС). Рассмотрены когнитивные аспекты информационного моделирования, которые выступают на первый план при проектировании и создании сложных ИС с привлечением большого коллектива интенсивно взаимодействующих между собой дистанционно удаленных специалистов. Особое внимание обращено на экономическую целесообразность детального и тщательного моделирования при проектировании и создании сложных и дорогостоящих систем. Все обоснования и выводы подкреплены примерами из практики проектирования и разработки сложных и дорогостоящих систем, в основном авиационных комплексов.

Ключевые слова: информационная модель; когнитивное моделирование; информационная система; проектирование и разработка

DOI: 10.14357/08696527160211

1 Введение

Проектирование современных ИС представляет собой исключительно сложный процесс, особенно в тех случаях, когда автоматизации в результате внедрения ИС подлежит сложная, многогранная и масштабная деятельность. Проведенные исследования [1, 2] позволили сделать вывод, что к таким сложным видам деятельности может быть отнесен процесс мониторинга и оценивания национальной безопасности, в частности национальной безопасности Российской Федерации. В этот процесс вовлечена большая часть российских министерств и ведомств, руководящий состав всех субъектов Федерации, а также ряд крупных компаний военно-промышленного комплекса, машиностроения и горнодобывающей отрасли. В процессе мониторинга и оценивания обобщению подлежат сотни разнородных качественных и количественных показателей, вычисление и нормализация которых осуществляются на основе десятков методик. В связи с этим к качеству проектирования предъявляются повышенные требования, так как последующая модификация недостаточно тщательно проработанного проекта ИС

¹ Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, gena-mslu@mail.ru

² Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, dmnik@a170.ipi.ac.ru

связана не только со значительными затратами финансовых, материальных и людских ресурсов, но и с временными потерями, что является критичным в системе государственного управления.

При создании ИС подобного уровня ограничено применение циклических методов проектирования, поэтому приходится исходить исключительно из «каскадного» метода разработки и создания ИС [3, 4]. Качество проекта и сдаваемой в промышленную эксплуатацию ИС зависит от ряда обстоятельств, однако немаловажным фактором становится информационное моделирование, на котором базируется разработка алгоритмов решения стоящих перед ИС задач и последующая реализация всей ИС [5, 6].

Настоящая статья посвящена анализу роли моделей в создании сложных ИС. Особое внимание уделяется экономической составляющей моделирования как обоснованному балансу между расходами на моделирование и степенью отражения в полученной модели свойств и характеристик моделируемой предметной области.

2 Роль информационной модели в проектировании информационной системы

С точки зрения проектирования ИС решение любой производственной или научной задачи в общем виде может быть представлено технологической цепочкой, показанной на рис. 1.

На этом рисунке вместо ИС представлена «Программа», так как с точки зрения логики решения прикладной задачи решающая роль в ИС принадлежит именно программному обеспечению. Вместе с тем необходимо отметить, что ИС как организационно-техническая система служит основой, на которой развертывается и функционирует программное обеспечение.

Кроме того, в этой цепочке важную роль играет звено «информационная модель» как необходимый и обязательный компонент ИС. Под информационной

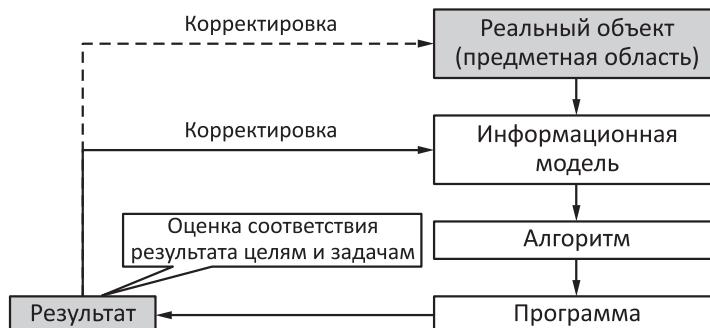


Рис. 1 Цикл проектирования

моделью понимается виртуальный образ предметной области, ее объектов или процессов, отражающий их существенные свойства с точки зрения целей и задач проектирования [7].

Рисунок 1 иллюстрирует несколько положений. Во-первых, качество информационной модели во многом определяет и качество создаваемого на ее основе программного обеспечения, а значит, и степень достижения целей проектирования ИС (основная цель — эффективное и экономически обоснованное решение стоящих перед ИС задач). Во-вторых, в ряде случаев после завершения разработки ИС возникает необходимость коррекции модели. Наконец, в-третьих, полученный результат в некоторых случаях позволяет принять решение о необходимости корректировки представлений как о производственных процессах, так и о самой онтологической модели предметной области, а также об уточнении целей и задач, решаемых ИС.

3 Основные подходы к моделированию

Чаще всего модель рассматривается как уменьшенная копия моделируемого объекта [8]. В некоторых случаях модель выполняется в масштабе один к одному к оригиналу. Так, при проектировании военно-транспортного самолета «Ан-12» конструкторами была создана его полноразмерная модель из дерева. Это позволило в деталях изучить варианты погрузки и выгрузки боевой техники и личного состава, а также особенности десантирования из самолета. Наконец, в ряде случаев о размерах модели нет смысла говорить вообще. Это характерно для моделирования абстрактных объектов и процессов, например процессов управлеченческой деятельности.

Некоторые авторы указывают [9], что модель отличается от оригинала материалом исполнения, что также неверно. В частности, для получения основных аэродинамических характеристик самолета его уменьшенную модель также изготавливают из дюралиюминия. Но даже в случае применения для модели другого материала, как это имело место в описанном выше случае с самолетом «Ан-12», авторы научных работ не раскрывают причину такого выбора. А ведь в данном случае речь идет прежде всего об экономически обоснованном решении — выборе более дешевого материала, обработка которого к тому же требует значительно меньших трудозатрат.

С другой стороны, авторы ряда научных работ подчеркивают [9], что модель — это упрощенное подобие реального объекта. Так, например, упрощенное моделирование при создании знаменитого пикирующего бомбардировщика «Пе-2» привело к тому, что самолет оказался недостаточно устойчивым по тангажу и рысканию, вследствие чего на нем было проблематично осуществлять бомбометание даже с горизонтального полета, не говоря уже о крутом пикировании.

В связи с этим возникает вопрос о необходимости упрощения. И во многих случаях дело здесь не в сложности объекта, на что обычно делается ссылка

в научных статьях, а в стоимости моделирования всего комплекса характеристик объекта. Однако шаблонные решения уместны не всегда и известны примеры, когда конструкторы и разработчики имеют возможность избежать упрощений. Примером этому может служить история проектирования советского стратегического бомбардировщика «Ту-4». Создание этой боевой машины осуществлялось методом «обратной разработки», т. е. копирования по образцу американского бомбардировщика «B-29». У команды А. Н. Туполева имелось в распоряжении три полнофункциональных самолета «B-29», один из которых был полностью разобран с целью изучения его узлов и агрегатов; второй был сохранен как оригинал, а третий самолет разработчики использовали в качестве летающей лаборатории. Наличие полнофункционального образца самолета «B-29» в качестве модели исключило необходимость создавать его упрощенную модель.

С учетом приведенных замечаний и комментариев можно предложить следующий вариант определения:

Модель — это имитация объектов и процессов предметной области, в необходимой степени отражающая ее важнейшие свойства и зависимости в контексте целей и задач моделирования, создаваемая с экономически обоснованными затратами материальных, финансовых, людских, временных и других видов ресурсов.

Для пояснения приведенной формулировки целесообразно вернуться к упомянутой выше проблеме упрощения: для изучения сложного объекта, и в частности исходя из принципа экономической целесообразности, прибегают к моделированию только необходимых и достаточных для целей и задач исследования характеристик, имея в виду возможные неточности и даже искажения.

Так, для изучения основных аэродинамических характеристик самолета можно пренебречь моделированием его силовой установки, систем управления вооружением или катапультирования, здесь достаточно уменьшенной твердотельной модели планера самолета. Однако в случае крыла изменяемой геометрии в этой модели необходимо предусмотреть «снятие» аэродинамических характеристик хотя бы в нескольких фиксированных положениях крыла: обязательно в крайних и в двух–трех промежуточных, т. е. и в данном случае необязательно моделировать всю систему управления крылом, так как она, во–первых, при исследовании аэродинамических характеристик не требуется, а во–вторых, аэродинамические характеристики для произвольного положения крыла можно довольно точно рассчитать методом интерполяции по полученным экспериментальным точкам.

4 Информационная модель

Из приведенных выше пояснений следует, что модель — это довольно широкое понятие, включающее в себя множество способов представления изучаемой предметной области. Границы между моделями различных типов или классов чаще всего условны. Наиболее распространены следующие признаки, по которым классифицируются модели [9]:

- цель применения;
- область знаний (предметная область);
- фактор времени;
- способ представления.

По целям использования выделяют модели учебные, опытные, имитационные, игровые, научно-технические. По области знаний модели могут быть биологическими, экономическими, историческими, социологическими и др. По фактору времени разделяют модели на динамические и статические: статическая модель описывает предметную область (объект, процесс) в фиксированный момент времени, т. е. дает «статический срез» предметной области; динамическая модель отражает процесс изменения или функционирования объекта или развития процесса во времени. По способу представления модели делятся на материальные (натурные) и идеальные (абстрактные).

Материальные модели — это вещественные образы, воспроизводящие внешние свойства, внутреннее строение либо действия моделируемых объектов. Этот вид моделирования характерен для экспериментального (опытного) метода познания. Например, уменьшенная в несколько раз и поэтому относительно дешевая натурная модель самолета позволяет проводить ее продувку в аэродинамической трубе и исследовать аэродинамические характеристики будущего летательного аппарата.

Идеальные модели неразрывно связаны с человеческим мышлением, с воображением и восприятием человеком окружающего мира. Абстрактное моделирование ориентировано на теоретический метод познания [10]. Абстрактные модели делятся на воображаемые и информационные.

Воображаемое (мысленное, или интуитивное) моделирование — это мысленное представление об объекте. Воображаемые модели формируются в воображении человека и сопутствуют его познавательной (когнитивной) деятельности.

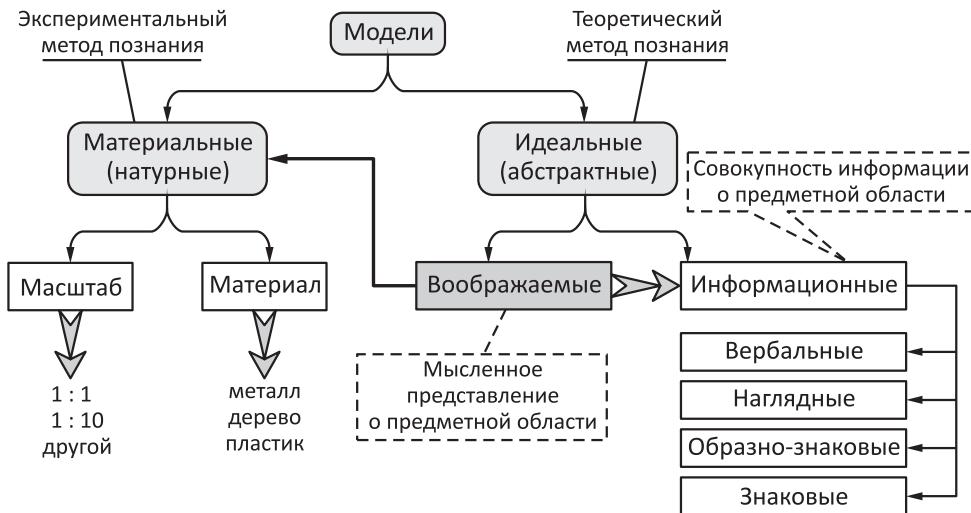
Информационная модель — это совокупность информации об объектах, процессах, их свойствах, взаимосвязях и отношениях, существующих в предметной области, используемой в контексте целей и задач проектирования и зафиксированной на материальном носителе.

Информационные модели представляют объекты в виде:

- словесных описаний, текстов;
- рисунков, эскизов, фотографий, кинофильмов;
- таблиц, схем, чертежей, карт;
- формул, логических выражений.

В зависимости от формы, способа представления и описания различают вербальные, наглядные, образно-знаковые и другие знаковые модели, как это изображено на рис. 2.

В вербальных моделях для отражения предметной области реального или абстрактного мира используются естественные языки. Форма представления

**Рис. 2** Способы представления моделей

такой модели — устное или письменное сообщение. Примерами служат учебные пособия и словари, инструкции по применению устройств и систем, регламенты обслуживания и эксплуатации, наставления и т. д.

В наглядных моделях свойства оригинала выражаются с помощью его образов: эскизов, рисунков, фотографий, кинофильмов. Сюда же относятся графические модели, в том числе геометрические фигуры и тела. Например, созданию натуруной модели самолета предшествует эскизное моделирование — разработка рисунков (эскизов) в нескольких проекциях, что позволяет еще до изготовления натуруной (материальной) модели мысленно исследовать объект, в данном случае — будущий самолет, согласовать с экспертами и с заказчиками его аэродинамическую форму и общую компоновку и внести коррективы в эскиз.

В образно-знаковых моделях применяются условные знаковые образы: схемы, графы, таблицы, чертежи, графики, планы, карты. Например: географическая карта, план квартиры, родословное дерево, блок-схема. К этой группе относятся структурные информационные модели, предназначенные для наглядного изображения структуры, составных частей и связей объектов.

Знаковые модели используют алфавиты формальных языков (условные знаки, специальные символы, буквы, цифры) и правила выполнения операций с элементами этих алфавитов. В качестве примеров можно назвать физические или химические формулы, математические выражения и формулы, нотную запись.

Одним из наиболее распространенных формальных языков в знаковом моделировании является алгебраический язык формул в математике, который

позволяет описывать функциональные зависимости между переменными и константами.

Математические модели основаны на формальных языках над конечными алфавитами, а изучаемое явление или процесс в них представлены в виде абстрактных объектов, выражающих:

- законы природы;
- внутренние свойства самих объектов;
- правила логических рассуждений.

Например, математическая модель качества крыла самолета¹ основывается на законах природы в области аэrodинамики, а математическая модель начисления заработной платы — на правилах, установленных в экономике.

Математическая модель — это часто система уравнений, описывающая физические, химические, биологические или иные процессы, принадлежащие той или иной предметной области. В качестве примера можно привести систему математических уравнений для вычисления дохода и прибыли предприятия в соответствии с некоторой принятой методикой.

В тех случаях, когда моделирование ориентировано на исследование моделей с помощью компьютера, одним из его этапов является разработка компьютерной модели.

Компьютерная модель — это виртуальный образ, созданный за счет вычислительных ресурсов, качественно и количественно отражающий внутренние свойства, внешние характеристики и связи моделируемого объекта в контексте целей и задач проектирования.

Компьютерная модель позволяет воспроизводить и анализировать внешний вид, строение или действие моделируемого объекта и функционирует посредством электромагнитных (реже оптических) сигналов. Разработка компьютерной модели предшествует созданию вербальных, наглядных образных и образно-знаковых моделей — все эти модели находят свое воплощение в виртуальном образе, формируемом в вычислительной среде.

Проектирование ИС основывается на идеальных (абстрактных) моделях, преимущественно образных и образно-знаковых. Вербальное моделирование ориентировано на разработку описаний, инструкций, регламентов, положений. В ряде случаев при создании ИС прибегают к наглядному моделированию, например, для отражения компоновки оборудования ИС или при решении эргономических задач. И все же главное предназначение информационных моделей для ИС заключается в описании информационных объектов и процессов в моделируемой предметной области.

¹Качество крыла определяется отношением его подъемной силы к лобовому сопротивлению. Чем выше этот показатель, тем лучше аэродинамические характеристики крыла. Вместе с тем качество крыла — это не постоянная величина: она зависит от скорости набегающего потока и угла атаки.

5 Когнитивное моделирование

Из структуры способов представления моделей, изображенной на рис. 2, следует вывод, что исходным этапом любого моделирования является формируемая в сознании человека воображаемая модель, предшествующая созданию как материальной, так и абстрактной (информационной) модели. Поэтому с учетом современных достижений когнитивной науки и психологии представляется целесообразным в рамках проблемы информационного моделирования рассмотреть ее когнитивную составляющую [11, 12]. Понятия когнитивной модели и когнитивного моделирования неразрывно связаны с сущностью информации как средства познания окружающего мира человеком.

В частности, для извлечения информации из данных, зафиксированных на материальном носителе (например, на листе бумаги) в форме текста, необходим целый комплекс методов доступа к этим данным: освещение, зрение, знание языка, на котором изложен материал, владение ключом шифрования (дешифрования), если текст зашифрован.

Например, информация может быть извлечена с листка бумаги путем его нагрева, если сообщение написано так называемыми симпатическими (невидимыми) чернилами, которые «проявляются» при нагреве или при обработке специальными химическими веществами. Можно привести еще более сложный пример, когда информация передается с помощью запаха духов (или одеколона) нанесенных на лист бумаги с нейтральным (ничего не значащим для получателя) текстом.

Роль и значение приведенных пояснений с точки зрения когнитивных методов и моделей вытекает из общей концепции моделирования, которое начинается с воображаемого моделирования, т. е. формирования обобщенной модели (представления) в сознании человека на основании некоторых исходных данных [13]. От адекватности этих представлений об объектах и структуре предметной области зависит результат последующего информационного моделирования и качество решения задач на основе этой модели. В частности, в образной модели человек не в состоянии воспроизвести все, даже наиболее существенные детали предметной области.

Во-первых, ситуация преднамеренно упрощается. Во-вторых, формирование образной модели при коллективном изучении проблемы зависит от способа интерпретации предметной области каждым членом коллектива. Наконец, в-третьих, человеку в принципе не свойственно мыслить в количественных терминах — он прежде всего мыслит качественными характеристиками [14]. Таким образом, разработка образной модели — это поиск общего замысла решения, которое в дальнейшем уже будет подробно «раскрыто» в информационной модели, поэтому когнитивная составляющая моделирования выступает на первый план.

Когнитивный подход при формировании образной модели в общем случае представлен на рис. 3. Представленный подход базируется на том положении, что картина мира включает в себя набор убеждений, особенностей восприятия, ценностных и практических установок, которыми человек руководствуется

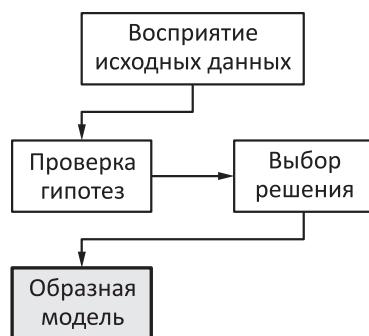


Рис. 3 Когнитивный подход при моделировании

результатам экспериментальных либо теоретических исследований уточняется воображаемая модель, а полученные изменения воплощаются в натурной либо абстрактной модели. Например, экспериментальные исследования аэродинамических характеристик уменьшенной модели летательного аппарата могут вынудить изменить представление о принципах работы и конфигурации крыла или фюзеляжа. На основании этих представлений вносятся изменения и в натурную модель, после чего проводится очередная итерация — серия экспериментальных исследований и последующий анализ их результатов.

в своей деятельности и которые влияют на процесс принятия решения, в данном случае — на формирование образной модели [15]. В процессе восприятия исходных данных и проверки гипотез каждый участник коллектива исходит из своих личных ценностных установок, но с учетом замечаний и предложений других членов.

Исходя из приведенных особенностей когнитивного моделирования, логику процесса моделирования можно представить в виде цепочки действий, изображенных на рис. 4.

Из представленной на рис. 4 логики процесса моделирования следует, что по ре-

зультатам экспериментальных либо теоретических исследований уточняется воображаемая модель, а полученные изменения воплощаются в натурной либо

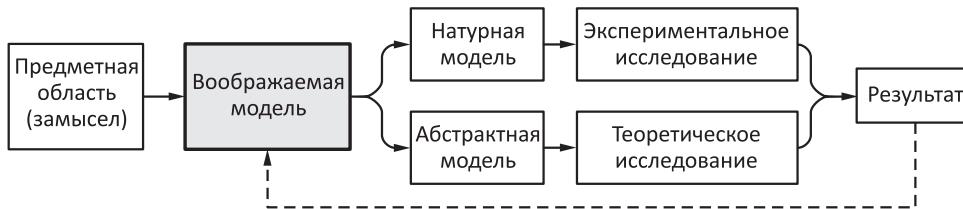


Рис. 4 Логика моделирования

6 Заключение

В настоящей работе рассмотрен традиционный цикл моделирования, где важнейшую роль играет этап создания информационной модели. Процесс информационного моделирования должен обеспечивать адекватность создаваемой модели целям и задачам автоматизируемой деятельности, экономическую целесообразность ее реализации, гибкость и масштабируемость модели и ИС в целом в условиях меняющихся целей и задач функционирования [16, 17]. Была рас-

смотрена специфика моделей, проведены параллели между классами моделей и их особенностями, а также решаемыми на их основе практическими задачами. Намечен подход к пониманию сущности и особенностей когнитивной составляющей информационного моделирования. Предложена схема (цепочка) этапов процесса когнитивного моделирования.

Литература

1. Лукьянов Г. В., Никишин Д. А., Веревкин Г. Ф., Косарик В. В. Нормативные и методологические аспекты организации информационного мониторинга национальной безопасности // Системы и средства информатики, 2015. Т. 25. № 3. С. 225–241.
2. Маслобоев А. В., Путилов В. А. Специфика и структура задачи информационной поддержки управления безопасностью региональных социально-экономических систем // Вестник МГТУ, 2015. Т. 18. № 3. С. 476–485.
3. Сергушин Г. С., Варламов О. О., Чибирова М. О., Елисеев Д. В., Муравьева Е. А. Информационное моделирование сложных автоматизированных систем управления технологическими процессами на основе миарных технологий // Искусственный интеллект, 2013. № 3. С. 126–138.
4. Сумин В. И., Колыхалин В. М., Смоленцева Т. Е. Описание функционирования информационного процесса сложных систем // Современные проблемы науки и образования, 2015. № 1. С. 97–102.
5. Репин В. В., Елиферов В. Г. Процессный подход к управлению. Моделирование бизнес-процессов. — М.: Манн, Иванов и Фербер, 2013. 544 с.
6. Васютинская С. И. Развитие информационного управления // Образовательные ресурсы и технологии, 2015. № 2(10). С. 113–119.
7. Ипатова Э., Ипатов Ю. Методологии и технологии системного проектирования информационных систем. — М.: Флинта, МПСИ, 2008. 215 с.
8. Городецкий А. Я. Информационные системы. Вероятностные модели и статистические решения. — СПб.: СПбГПУ, 2003. 326 с.
9. Артёмова С. В. Информатика. — Тамбов: ТГТУ, 2001. Ч. I. 160 с.
10. Ozherel'eva T. A. Resource information models // Perspect. Sci. Educ., 2015. Vol. 13. Iss. 1. P. 39–44.
11. Авдеева З. К., Коврига С. В., Макаренко Д. И. Когнитивное моделирование для решения задач управления слабоструктурированными системами (ситуациями) // Управление большими системами. — М.: ИПУ РАН, 2007. Вып. 16. С. 26–39.
12. Авдеева З. К., Коврига С. В. Формирование стратегии развития социально-экономических объектов на основе когнитивных карт. — Saarbrucken: LAP LAMBERT Academic Publs., 2011. 184 с.
13. Tsvetkov V. Ya. Cognitive science of information retrieval // Eur. J. Psychol. Stud., 2015. Vol. 5. Iss. 1. P. 37–44.
14. Авдеева З. К., Коврига С. В., Макаренко Д. И., Максимов В. И. Когнитивный подход в управлении // Contr. Sci., 2007. № 3. С. 2–8.
15. Tsvetkov V. Ya. Information situation and information position as a management tool // Eur. Researcher, 2012. Vol. 36. No. 12-1. P. 2166–2170.

16. Потапов А. Н., Абъ-Абед Ф. Н., Мартинов Д. В., Кордюков Р. Ю. Объективные методы вероятностной оценки степени адекватности имитационного моделирования в информационных системах освоения эргатических радиоэлектронных объектов // Программные продукты и системы, 2015. № 3(111). С. 97–103.
17. Баранюк Т. В. Методика временного анализа информационных систем // Образовательные ресурсы и технологии, 2015. № 1(9). С. 108–113.

Поступила в редакцию 11.01.16

COGNITIVE FEATURES OF INFORMATION MODELING DURING DESIGN OF COMPLEX INFORMATION SYSTEMS

G. V. Lukyanov and D. A. Nikishin

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: The article is devoted to the problems of information modeling of complex information systems. Two of the most controversial and debatable issues of information modeling are considered: the business case for simulation and its cognitive aspects. The article shows that the “driving” force of modeling is the economic feasibility of this activity in the process of designing and creating complex and expensive systems. Cognitive aspects come to the fore during design and creation of complex information systems involving a large team of intensively interacting specialists. All studies and findings are backed up with convincing examples from the practice of design and development of complex and expensive systems, mainly, aircraft systems.

Keywords: information model; cognitive modeling; information system; design and development

DOI: 10.14357/08696527160211

References

1. Luk'yanov, G. V., D. A. Nikishin, G. F. Verevkin, and V. V. Kosarik. 2015. Normativnye i metodologicheskie aspekty organizatsii informatsionnogo monitoringa na-tional'noy bezopasnosti [Standard and methodological aspects of the organization of information monitoring of national security]. *Sistemy i Sredstva Informatiki—Systems and Means of Informatics* 25(3):225–241.
2. Masloboev, A. V., and V. A. Putilov. 2015. Spetsifika i struktura zadachi informatsionnoy podderzhki upravleniya bezopasnost'yu regional'nykh sotsial'no-ekonomiceskikh sistem [Specific features and structure of a problem of information support of management of safety of regional social and economic systems]. *Vestnik MGTU* [Bulletin of MGTU] 18(3):476–485.

3. Sergushin, G. S., O. O. Varlamov, M. O. Chibirova, D. V. Eliseev, and E. A. Murav'eva. 2013. Informatsionnoe modelirovaniye slozhnykh avtomatizirovannykh sistem upravleniya tekhnologicheskimi protsessami na osnove mivarnykh tekhnologiy [Information modeling of difficult automated process control systems on a basis of the mivarnykh technologies]. *Iskusstvennyy Intellekt* [Artificial Intelligence] 3:126–138.
4. Sumin, V. I., V. M. Kolykhalin, and T. E. Smolentseva. 2015. Opisanie funkcionirovaniya informatsionnogo protsessa slozhnykh sistem [Description of functioning of information process of difficult systems]. *Sovremennye Problemy Nauki i Obrazovaniya* [Modern Problems of Science and Education] 1:97–102.
5. Repin, V. V., and V. G. Eliferov. 2013. *Protsessnyy podkhod k upravleniyu. Modelirovaniye biznes-protsessov* [Process approach to management. Modeling of business processes]. Moscow: Mann, Ivanov & Ferber. 544 p.
6. Vasyutinskaya, S. I. 2015. Razvitiye informatsionnogo upravleniya [Development of information management]. *Obrazovatel'nye Resursy i Tekhnologii* [Educational Resources and Technologies] 2(10):113–119.
7. Ipatova, E., and Yu. Ipatov. 2008. *Metodologii i tekhnologii sistemy proektirovaniya informatsionnykh sistem* [Methodologies and technologies of system design of information systems]. Moscow: Flinta, MPSI. 215 p.
8. Gorodetskiy, A. Ya. 2003. *Informatsionnye sistemy. Veroyatnostnye modeli i statisticheskie resheniya* [Information systems. Probabilistic models and statistical decisions]. SPb: Izd-vo SPbGPU [SPbGPU Publs.]. 326 p.
9. Artemova, S. V. 2001. *Informatika*. [Informatics]. Tambov: Izd-vo TGTU [TSTU Publs.]. Part I. 160 p.
10. Ozherel'eva, T. A. 2015. Resource information models. *Perspect. Sci. Educ.* 13(1):39–44.
11. Avdeeva, Z. K., S. V. Kovriga, and D. I. Makarenko. 2007. Kognitivnoe modelirovaniye dlya resheniya zadach upravleniya slabostrukturirovannymi sistemami (situatsiyami) [Cognitive modeling for the solution of problems of management of semistructured systems (situations)]. *Upravlenie Bol'shimi Sistemami* [Management of Big Systems]. Moscow: IPU RAN. 16:26–39.
12. Avdeeva, Z. K., and S. V. Kovriga. 2011. *Formirovaniye strategii razvitiya sotsial'no-ekonomicheskikh ob'ektov na osnove kognitivnykh kart* [Formation of strategy of development of social and economic objects on the basis of cognitive maps]. Saarbrucken: LAP LAMBERT Academic Publs. 184 p.
13. Tsvetkov, V. Ya. 2015. Cognitive science of information retrieval. *Eur. J. Psychol. Stud.* 5(1):37–44.
14. Avdeeva, Z. K., S. V. Kovriga, D. I. Makarenko, and V. I. Maksimov. 2007. Kognitivnyy podkhod v upravlenii [Cognitive approach in management]. *Contr. Sci.* 3:2–8.
15. Tsvetkov, V. Ya. 2012. Information situation and information position as a management tool. *Eur. Researcher* 36(12-1):2166–2170.
16. Potapov, A. N., F. N. Abu-Abed, D. V. Martynov, and R. Yu. Kordyukov. 2015. Ob'ektivnye metody veroyatnostnoy otsenki stepeni adekvatnosti imitatsionnogo modelirovaniya v informatsionnykh sistemakh osvoeniya ergaticheskikh radioelektronnykh ob'ektov [Objective methods of a probabilistic assessment of degree of adequacy of imitating modeling in information systems of development of ergatic radioelectronic objects]. *Programmnye Produkty i Sistemy* [Software and Systems] 3(111):97–103.

17. Baranyuk, T. V. 2015. Metodika vremennogo analiza informatsionnykh sistem [Technique of the temporary analysis of information systems]. *Obrazovatel'nye Resursy i Tekhnologii* [Educational Resources and Technologies] 1(9):108–113.

Received January 11, 2016

Contributors

Lukyanov Gennady V. (b. 1952)— Candidate of Military Science (PhD), associate professor; Head of Laboratory, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation, gena-mslu@mail.ru

Nikishin Dmitry A. (b. 1976) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; dmnikishin@mail.ru

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ОБРАЗОВАНИЕ В ЕВРОСОЮЗЕ: ФОРМИРОВАНИЕ НОВЫХ ПРОФЕССИОНАЛОВ

Д. А. Богданова¹

Аннотация: Рассматривается ситуация несоответствия на рынке труда, существующая, несмотря на высокий уровень безработицы в Европейском Союзе, между спросом на специалистов по информационно-коммуникационным технологиям (ИКТ) и предложением. Анализируются причины существующего дисбаланса, вызванного в основном недостаточным уровнем сформированных цифровых компетенций у значительной части трудоспособного населения. Приводятся отдельные положения рижской декларации «e-Skills for Jobs», разработанной для изменения существующей ситуации. Значительная роль в компенсации дисбаланса отводится школьным учителям, которые, как показывают результаты проводимых исследований, недостаточно используют или избегают использования ИКТ в учебном процессе. Перечисляются меры, предпринимаемые для повышения квалификации учителей, например массовые открытые онлайн-курсы, тематика которых ориентирована на различные аспекты использования ИКТ в преподавании как общеобразовательных дисциплин, так и информатики.

Ключевые слова: цифровые компетенции; цифровые технологии; ИКТ; повышение квалификации; обучение учителей; массовые открытые онлайн-курсы; безопасность детей в сети Интернет; самооценивание; робототехника; программирование

DOI: 10.14357/08696527160212

Несмотря на высокий уровень безработицы, в Европе ощущается все возрастающая нехватка специалистов в области ИКТ. Тысячи объявлений о новых рабочих местах остаются невостребованными. В государствах Евросоюза очень высок уровень безработицы среди молодежи возраста 15–24 лет, а зарплаты существенно ниже, чем в США или странах Азиатско-Тихоокеанского региона. Специалисты говорят о необходимости увеличения числа людей, обладающих соответствующими профессиональными навыками. Это особенно справедливо сегодня, поскольку ИКТ трансформируют практически каждую область экономической жизни. Задача обучения необходимым навыкам европейского рынка рабочей силы стала проблемой номер один в последнее время в связи с возрастающей информатизацией единого европейского пространства.

¹Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, d.a.bogdanova@mail.ru

В марте 2015 г. в Риге стартовала программа, рассчитанная на 2015–2016 гг., под названием «The Riga Declaration on e-Skills for Jobs» — рижская декларация о цифровых (электронных) умениях для работы, ставящая своей целью привлечь внимание стран-участниц к растущей проблеме [1]. Разработчики программы уверены, что цифровые технологии откроют мир для европейского бизнеса и Европу для мирового рынка, стимулируя европейских производителей к повышению конкурентоспособности. В рамках рижской Декларации был разработан перечень из сорока цифровых компетенций «The e-Competence Framework» для будущих соискателей работы в области ИКТ, включающий и разработанный ранее перечень основных умений для выпускников школ и колледжей [2]. По мнению разработчиков, молодые люди должны будут уметь организовывать «бэкап» в облаках, редактировать на базовом уровне фото и видео, конвертировать форматы файлов, пользоваться банковскими онлайн-услугами, брендировать себя в цифровом сообществе, знать Google docs, Microsoft Office, HTML и владеть базовыми навыками программирования.

Переход от базовых цифровых компетенций к расширенным навыкам должен происходить поэтапно, на основании реализации целостного подхода, сочетающего неформальное и формальное преподавание, объединение академических знаний и практического опыта, необходимых для работы и жизни в цифровом мире. Программа предполагает овладение в перспективе и такими продвинутыми ИКТ, как мобильные коммуникации, социальные медиа, облачные технологии, умение работать с big data и Интернетом вещей. Очевидно, что одной из важнейших, первоочередных задач в рамках реализации положений программы является совершенствование школьных учебных программ за счет включения в них информатики и программирования, а также расширение информационной инфраструктуры [1].

Однако для эффективной реализации программы современные учителя должны быть хорошо подготовлены к преподаванию с использованием современных ИКТ. Эта проблема актуальна для многих европейских государств. В большинстве государств Евросоюза, например, умение использовать технологии и знание того, каким образом они могут обеспечить современный уровень преподавания, пока еще не стало неотъемлемой составляющей требований к профессиональным умениям сегодняшнего учителя. Для решения этой задачи в Евросоюзе регулярно предпринимаются серьезные шаги. В частности, для общеевропейского образовательного портала уже 6 лет назад, практически сразу после его запуска, был разработан 21 автономный информационный модуль, обеспечивающий возможности для самообучения в зависимости от первоначального уровня ИКТ-компетентности учителя и его опыта работы с метаданными и хранилищами [3]. Предполагалось, что в результате прохождения обучения учителя смогут: осуществлять поиск образовательных ресурсов на различных порталах; определять, как ресурс может быть использован; адаптировать его к своим целям; обмениваться образовательным контентом через социальные сети и т. д. На протяжении последнего времени Еврокомиссия в рамках реализации программы «Digital

agenda for Europe 2020», стартовавшей в 2010 г., один раз в два года организует обучение школьных учителей использованию ИКТ на уроках.

Однако исследования показывают, что уровень использования ИКТ в учебном процессе по-прежнему отстает от темпов развития самих средств. Исследователи выдвигают разные объяснения: от неуверенности в своих силах до комплекса причин, включающих доступность ИКТ, а также компетентность, мотивацию и иные объективные и личностные факторы.

Выступая на конференции BETT-2015, В. Филипс, директор по образованию Фонда Билла и Мелинды Гейтс, предложила свое видение проблемы. Она считает, что продавцы технологий больше заинтересованы в продуктах, с которыми им самим, а не потенциальным пользователям, будет проще иметь дело. Когда же дело доходит до школьной администрации, то она в первую очередь смотрит на сумму, которую придется заплатить. Таким образом, ни один из элементов цепочки производитель–потребитель не предусматривает учета реальных потребностей учителя [4]. И это, по ее мнению, гораздо более серьезный вопрос, чем может показаться на первый взгляд. При таком подходе энтузиасты теряют веру и отказываются от проектов. «Технологии сами по себе не могут установить связь с учеником, не могут руководить проектом, не могут выяснить причину, по которой у него не идут дела». Технологии могли бы сделать учителя ближе к своим учащимся, но в настоящее время покупателем является не учитель. Признавая справедливость сказанного, тем не менее хочется отметить, что для того, чтобы сформулировать свои потребности, учитель должен быть компетентным, уверененным пользователем ИКТ.

Неуверенность в своих силах в определенной степени укладывается в понятие «технологический разрыв поколений», когда учащиеся или молодые учителя являются более уверенными и активными пользователями ИКТ. В конце августа 2014 г. Министерство связи Великобритании опубликовало результаты своего ежегодного маркетингового исследования. Было опрошено 2000 взрослых и 800 детей. В исследовании был использован новый показатель: коэффициент цифровой уверенности (digital confidence quotient). Он показывает уровень пользовательских умений во взаимодействиях с цифровыми технологиями. Оказалось, что самыми уверенными пользователями являются подростки возраста 14–15 лет. Дети 6-летнего возраста по уровню умений сравнялись с 45-летними взрослыми [5].

Результаты последних исследований Еврокомиссии, в которых приняло участие более 190 000 учителей, учащихся и директоров школ из 31 страны (27 государств Евросоюза, а также Хорватии, Исландии, Норвегии и Турции), проведенных в 2011–2012 гг., были опубликованы в 2013 г. [6]. В среднем по Евросоюзу учащиеся ответили, что у них бывают занятия с использованием ИКТ от одного до нескольких раз в месяц. Часть опрошенных учителей ответили, что использование ИКТ не приносит какой-либо ощутимой пользы на уроках. Очевидно, что по сравнению с предыдущим исследованием [7] учителя более подготовлены, признают возможное положительное влияние использования ИКТ в учебном

процессе на учащихся. Обучение учителей использованию ИКТ в учебном процессе в редких странах является обязательным. Не более 25%–30% учащихся обучают те учителя, для которых подобная подготовка была обязательной.

Исследование в целом показало, что как учащиеся, так и учителя часто используют ИКТ в различных учебных занятиях. Отмечена высокая мотивация учащихся и хорошие результаты в тех школах, где в рамках школьной внутренней политики существуют определенные требования по внедрению ИКТ и поощрение учителей за использование ИКТ, сочетающиеся с поддержкой со стороны администрации в плане повышения квалификации. Директора таких школ практически единогласно говорят о необходимости использования ИКТ в учебном процессе, так как этот аспект является очень важным в подготовке учащихся к работе и жизни в XXI в., а подавляющее большинство учащихся говорят о том, что использование ИКТ на уроке меняет общую атмосферу и качественно изменяет в лучшую сторону процесс обучения. В среднем по Евросоюзу примерно 20%–25% учащихся учатся у ИКТ-компетентных учителей, имеющих доступ к технологиям и не испытывающих затруднений в их использовании. Более подробные результаты дают такую картину: от 30% до 50% учащихся 4–8 классов обучают ИКТ-компетентные учителя в Болгарии, Эстонии, Ирландии, Португалии, Словакии, Словении и Швеции. И наоборот, менее 10% таких учителей преподают в школах Австрии, Бельгии, Кипра, Франции, Греции и Люксембурга. При этом результаты опросов учащихся зачастую говорят о том, что компетентные учителя чаще используют ИКТ на уроках, несмотря на ограниченный доступ и существующие препятствия. И, несмотря на хорошее оснащение школ, реже прибегают к использованию ИКТ те учителя, которые не чувствуют себя уверенными пользователями.

Средний уровень технической оснащенности школ по Евросоюзу таков: 3–7 учащихся на один компьютер, при этом чем старше учащиеся, тем ниже этот показатель в большинстве стран. В среднем более 100 учащихся приходится на одну интерактивную доску и 50 — на проектор. Более 90% учащихся имеют в школах широкополосное подключение к Интернету со скоростью 2–30 Мб/с. Большинство школ имеют базовое подключение: поддерживают сайт, электронную почту для учащихся и учителей или виртуальную обучающую среду.

Отмечается, что компьютеров в школах стало в 2 раза больше, чем было в 2006 г., когда проводилось аналогичное исследование, но на основе другой методики [7]. Однако данные по странам очень сильно отличаются. Отмечена тенденция к использованию компьютеров меньшего размера по сравнению с 2006 г. и увеличение количества личных устройств, например мобильных телефонов. Число учителей, использующих компьютер для подготовки урока, практически удвоилось и выросло до почти 90% с 2006 г., однако количество учителей, использующих ИКТ на более чем 25% своих уроков, практически не изменилось с 2006 г.

Учащиеся имеют более продолжительный опыт занятий с компьютером дома, чем в школе. Примерно 80%–90% учащихся имеют более чем четырехлетний

опыт пользования домашним компьютером по сравнению с 40%–60% с таким же показателем в школе. У исследователей вызвал беспокойство низкий уровень использования цифровых образовательных ресурсов и электронных учебников. При этом только порядка 30% учащихся используют электронные учебники и мультимедийный инструментарий от одного до нескольких раз в неделю, в то время как 50% учащихся всех возрастов никогда не пользуются такими ресурсами. Несмотря на различие в методиках исследований, результаты 2006–2009 и 2011–2013 гг. возможно сравнивать. В условиях заметного роста числа компьютеров в школах исследователи, тем не менее, не выявили корреляции между оснащенностью школы компьютерами и частотой их использования учителями и учащимися. Очевидно, что многое по-прежнему зависит от личности учителя.

И здесь хочется отметить, что среди специалистов, особенно в последние годы, большой популярностью пользуется метафора карандаша — «pencil metaphor», отражающая разновидности реакции учителей на внедрение ИКТ [8]. Год появления этой метафоры не известен, однако в 2006 г. на конференции ISTE (International Society Technology for Education) она уже была в ходу, а в 2012 г. эта метафора была использована для презентации на конференции в Южной Африке. Нет указания и на результаты исследований, которые бы проводились с использованием этой метафоры. Это всего лишь метафора. Но она довольно часто упоминается, когда речь идет об учителях и использовании ИКТ. Карандаш условно делится на 6 частей: 1 — заточенный грифель; 2 — часть заточенного деревянного корпуса; 3 — деревянный корпус; 4 — обод наконечника; 5 — наконечник; 6 — ластик. Категория, относящаяся к первой группе, — это новаторы, готовые принять и адаптировать технологии в своей работе. Они с энтузиазмом готовы делиться накопленным опытом. И, подобно грифелю, некоторые из них довольно быстро теряют энтузиазм и нуждаются в «перезаточке». Представители второй категории наблюдают за действиями первооткрывателей, заимствуют и применяют достижения в своей практике, учитывая ошибки предшественников, и достигают значительных успехов в своей педагогической деятельности. Третья группа — это деревянный корпус. Эта категория готова принять и использовать технологии в том случае, если кто-то получит и установит необходимое оборудование, обучит их и будет присматривать, чтобы всё и все функционировали. Если их чему-нибудь научит человек из первой группы, они это примут и будут действовать, как обучены. Следующая часть — это обод наконечника, а в иной модели карандаша — это часть деревянного корпуса, которая никогда не станет острой, каким бы маленьким ни становился карандаш. Эта категория строго придерживается того, что знает, является сторонником традиционных методов обучения и считает, что во внедрении технологий в их практику нет никакой необходимости. Следующая категория — наконечник. Эта группа посещает все семинары, все занятия, всегда все конспектирует, всегда произносит правильные слова, но никогда ничего не делает. И, наконец, ластик. По своей природе люди этой группы являются противниками новшеств и стараются отменить, аннулировать то, что было сделано энтузиастами.

По мнению специалистов, это удачная аллегория, использующая образ карандаша как одного из первых инструментов в обучении.

Сознавая важность стоящих задач и необходимость повышения уровня цифровых компетенций учителей, ответственные лица в Евросоюзе запустили массовые открытые онлайн-курсы на платформе собственной разработки European Schoolnet Academy. К моменту написания статьи уже было проведено более 10 курсов, посвященных вопросам использования ИКТ в учебном процессе (два из них автор успешно окончил). Тематика охватывает самые разные аспекты: от использования игр в обучении до творческого использования планшетов на уроках, от робототехники и обучения программированию до возможных сценариев обучения в будущем. Эта часть занятий ориентирована на варианты использования ИКТ на уроках. Особые надежды связаны с обучением программированию как одним из путей профилактики ошибочного поведения детей в сетях. Сторонники такого похода считают, что, если дети будут знать, как Всемирная паутина устроена изнутри, они будут более благоразумны в своем сетевом поведении. Значительное внимание уделяется помощи учителям в освоении ИКТ. Здесь имеется в виду, что, как правило, учителя осваивают новые навыки и приобретают дополнительные знания за счет своего личного времени, что особо отмечается организаторами курсов в информационных материалах. Поэтому каждый курс, со средней продолжительностью 6–7 нед., спроектирован таким образом, чтобы занять не более трех часов личного учительского времени в неделю. И помимо методических аспектов учителей знакомят с новым разработанным инструментарием, таким как, например, Learning Designer, разработка London Knowledge Lab — Institute of Education, программа-помощник в подготовке планов уроков.

Следует особо отметить, что в последнее время особое внимание в Европе уделяется технологиям самооценивания. И этому направлению (Mentoring technology enhanced pedagogy) будет посвящен целый цикл курсов. Первый из них уже состоялся, и это был один из упомянутых ранее двух курсов, на которых автор прошел обучение. В момент проведения занятий слушателям курса была предложена программа-тест норвежской разработки, позволяющая оценить уровень собственной ИКТ-компетентности и получить на основании сформировавшейся оценки запрограммированный совет, на что следует обратить внимание. Наряду с самооцениванием также довольно широко продвигается взаимное оценивание коллег.

Что касается применений ИКТ на уроках, предлагаемых слушателям курсов для ознакомления, здесь палитра демонстрируемых средств и методик достаточно широкая. Предлагаются примеры самых разнообразных современных учительских практик: это и танцевальный коврик, и интерактивная доска, и платы с микроконтроллерами, и робототехника, и среды программирования, и социальные сети. Кроме того, учителям дается обзор технологических изменений в вопросах найма (интервью, резюме, портфолио), организации рабочих мест и рабочих коллективов (дистанционная работа), способов достижения результа-

тов и т. д. Полный анализ содержания и методов организации занятий, в которых принял участие автор, может быть предметом отдельного материала.

Цифровые умения останутся на повестке дня в ближайшей и более отдаленной экономической перспективе. Эта тема остро стоит не только в Евросоюзе, но и в Соединенных Штатах Америки. Произошел пересмотр учебных программ в сторону обучения программированию в школах, а лозунг «*No child left behind*» («Ни один ребенок не забыт») был заменен новым: «*Every student succeeds*» («Каждый ученик преуспевает»). Компания Google второй год проводит летние школы по обучению программированию. Академия Хана проводит международный программистский флешмоб «Час программирования», в котором в прошлом году приняли участие и российские школы. Президент Обама недавно обратился к Конгрессу с просьбой о дополнительном выделении 4 млрд долларов на перестройку системы образования с целью введения в школьные образовательные программы штатов обучения программированию [9]. Проблема с дефицитом подготовленных в области информационных технологий специалистов наметилась и в России — и для ее решения также начали предприниматься определенные шаги: увеличено число бесплатных мест при приеме в вузы на специальности, связанные с программированием; расширяется методология организации и количество проводимых олимпиад для школьников по программированию. Цифровые умения будут способствовать повышению производительности в результате использования цифровых технологий и приведут в соответствие растущие потребности промышленности в квалифицированном персонале за счет расширения возможностей для каждого отдельного человека в сложных условиях, существующих на европейском рынке труда. Хочется верить, что столь серьезный комплексный подход к решению проблем формирования и повышения уровня цифровой грамотности трудоспособного населения и подрастающего поколения в государствах Евросоюза принесет свои плоды. А еще очень хочется, чтобы и Россия интенсифицировала и расширила спектр принимаемых мер, чтобы не растерять, а сохранить свои позиции на фоне столь серьезных усилий, принимаемых в Европе и в мире. Таким образом, следует констатировать, что жизнь доказывает справедливость тезиса академика А. П. Ершова: «Программирование — вторая грамотность».

Литература

1. The Riga Declaration on e-skills. <http://eskillsforjobs.lv/wp-content/uploads/2015/03/The-Riga-Declaration-on-e-Skills-13-March-2015.pdf>.
2. Petronzio M. Don't leave college without these 10 digital skills // Mashable, 06.05.2013. <http://mashable.com/2013/05/06/digital-skills-college/#bdGuzUTwsqg>.
3. Богданова Д. А. О методической поддержке образовательного портала Евросоюза // Новые информационные технологии в образовании (НИТО-2011): Мат-лы Междунар. науч.-практич. конф. — Екатеринбург: РГППУ, 2011. Ч. 2. <http://nito.rsvpu.ru/files/nito2011/nito2011-2.pdf>.

4. Bett show 2015. <http://seen.co/event/bett-show-2015-london-2015-3555/highlight/438548>.
5. Богданова Д. А., Березина Н. Л. Опыт формирования цифровой и медиаграмотности: на пути к цифровому гражданству // Новые вызовы для педагогики и качества образования: массовые открытые онлайн-курсы, облачные сервисы, мобильные технологии (ИИТО-2014): Мат-лы Междунар. конф. ЮНЕСКО. — Москва, 2014. 5 с. <http://conference2014.iite.unesco.org/wp-content/uploads/2014/11/Bogdanova.pdf>.
6. Survey of schools: ICT in education. Benchmarking access, use and attitudes to technology in Europe's schools. — European Union, 2013. Final Study Report. <https://ec.europa.eu/digital-agenda/sites/digital-agenda/files/KK-31-13-401-EN-N.pdf>.
7. Богданова Д. А. О модели, используемой для анализа склонности учителей к использованию информационно-коммуникационных технологий в учебном процессе // Информационные технологии для Новой школы: Мат-лы Всеросс. конф. — СПб.: РЦОКОИТ, 2011. С. 29–31.
8. 6 ways teachers respond to education technology // TeachThought, 28.08.2014. <http://www.teachthought.com/technology/pencil-metaphor-how-teachers-respond-to-education-technology>.
9. Brown E. Obama outlines \$4 billion ‘Computer Science for All’ education plan // The Washington Post, 30.01.2016. https://www.washingtonpost.com/local/education/obama-outlines-4-billion-computer-science-for-all-education-plan/2016/01/29/3ad40da2-c6d9-11e5-9693-933a4d31bcc8_story.html.

Поступила в редакцию 16.03.16

INFORMATION TECHNOLOGIES AND EDUCATION IN THE EUROPEAN UNION: FORMING THE NEW PROFESSIONALS

D. A. Bogdanova

Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation

Abstract: The situation in the labour market mismatch between the availability and demand for information and communications technologies specialists that exists in the European Union despite the high level of unemployment is considered. The reasons for the existing imbalance that has mainly been caused by insufficient level of digital competences among the significant part of the working population are analyzed. Certain aspects of “The Riga Declaration on e-skills” designed to change the current situation are presented. A significant role in the imbalance compensation is assigned to school teachers, who either insufficiently use or avoid the ICT usage in the learning process, which has been proved by the

research results. The measures taken to improve the teachers qualification level are listed. The Massive Open Online Courses focused on various aspects of the ICT use in teaching both general subjects and computer science are given as an example.

Keywords: digital competences; digital technologies; professional development; training teachers; Massive Open Online Courses; kids digital safety; self-assessment; robotics; coding

DOI: 10.14357/08696527160212

References

1. The Riga Declaration on e-skills. Available at: <http://eskillsforjobs.lv/wp-content/uploads/2015/03/The-Riga-Declaration-on-e-Skills-13-March-2015.pdf> (accessed January 31, 2016).
2. Petronzio, M. May 6, 2013. Don't leave college without these 10 digital skills. *Mashable*. Available at: <http://mashable.com/2013/05/06/digital-skills-college/#bdGuzU.Twsqg> (accessed January 31, 2016).
3. Bogdanova, D. A. 2011. O metodicheskoy podderzhke obrazovatel'nogo portala Evrosoyuza [On the methodological support of the educational portal of the European Union]. *New Information Technologies in Education: Scientific-Practical Conference (International) Proceedings*. Part 2. Available at: <http://nito.rspu.ru/files/nito2011/nito2011-2.pdf> (accessed January 31, 2016).
4. Bett show 2015. Available at: <http://seen.co/event/bett-show-2015-london-2015-3555/highlight/438548> (accessed January 31, 2016).
5. Bogdanova, D. A., and N. L. Berezina. Opty formirovaniya tsifrovoy i mediagramotnosti: Na puti k tsifrovomu grazhdanstvu [Shaping science and media literacy: On the way to digital citizenship]. *IITE 2014: Conference (International) "New Challenges for Pedagogy and Quality Education: MOOCs, Clouds and Mobiles" Proceedings*. Available at: <http://conference2014.iite.unesco.org/wpcontent/uploads/2014/11/Bogdanova.pdf> (accessed January 31, 2016).
6. European Union. 2013. Survey of schools: ICT in education. Benchmarking access, use and attitudes to technology in Europe's schools. Final Study Report. Available at: <https://ec.europa.eu/digital-agenda/sites/digital-agenda/files/KK-31-13-401-EN-N.pdf> (accessed January 31, 2016).
7. Bogdanova, D. A. 2011. O modeli, ispol'zuemoy dlya analiza sklonnosti uchiteley k ispol'zovaniyu informatsionno-kommunikatsionnykh tekhnologiy v uchebnom protsesse [About the model, describing propensity to the use of computers and internet by teachers in classroom situations at schools]. *Mat-ly Vseross. konf. "ITO Sankt-Peterburg. Informatsionnye tekhnologii dlya novoy shkoly"* [Conference "ITO St. Petersburg. Information technology for the new school" Proceedings]. 29–31. Available at: <http://rckoit.ru/dld/cjnf/itosbornik.pdf> (accessed: January 31, 2016).
8. 6 ways teachers respond to education technology. August 28, 2014. *TeachThought*. Available at: <http://www.teachthought.com/technology/pencil-metaphor-how-teachers-respond-to-education-technology> (accessed January 31, 2016).

9. Brown, E. January 30, 2016. Obama outlines \$4 billion ‘Computer Science for All’ education plan. *The Washington Post*. Available at: https://www.washingtonpost.com/local/education/obama-outlines-4-billion-computer-science-for-all-education-plan/2016/01/29/3ad40da2-c6d9-11e5-9693-933a4d31bcc8_story.html (accessed January 31, 2016).

Received March 16, 2016

Contributor

Bogdanova Diana A. (b. 1946) — Candidate of Science (PhD) in education, senior scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44-2 Vavilova Str., Moscow 119333, Russian Federation, d.a.bogdanova@mail.ru

ОБ АВТОРАХ

Адамович Игорь Михайлович (р. 1934) — кандидат технических наук, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Анисимов Владимир Иванович (р. 1926) — доктор технических наук, профессор Санкт-Петербургского государственного электротехнического университета «ЛЭТИ»; главный научный сотрудник Центра информационных технологий в проектировании Российской академии наук (ЦИТП РАН)

Бахтеев Олег Юрьевич (р. 1993) — студент Московского физико-технического института

Богданова Диана Александровна (р. 1946) — кандидат педагогических наук, старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Васильев Сергей Алексеевич (р. 1991) — аспирант Санкт-Петербургского государственного электротехнического университета «ЛЭТИ»

Волков Олег Игоревич (р. 1964) — ведущий программист Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Гасилов Артур Владимирович (р. 1992) — инженер-исследователь Орловского филиала Федерального исследовательского центра «Информатика и управление» Российской академии наук

Григорьев Семён Вячеславович (р. 1989) — аспирант Санкт-Петербургского государственного университета

Гридин Владимир Николаевич (р. 1944) — доктор технических наук, научный руководитель Центра информационных технологий в проектировании Российской академии наук (ЦИТП РАН)

Дударов Сергей Павлович (р. 1977) — кандидат технических наук, доцент, декан факультета информационных технологий и управления Российского химико-технологического университета им. Д. И. Менделеева

Ильин Владимир Дмитриевич (р. 1937) — доктор технических наук, заведующий лабораторией Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Корепанов Эдуард Рудольфович (р. 1966) — кандидат технических наук, заведующий сектором Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Лукьянов Геннадий Викторович (р. 1952) — кандидат военных наук, доцент; заведующий сектором Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Никишин Дмитрий Александрович (р. 1976) — кандидат технических наук, заведующий сектором Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Папаев Павел Леонидович (р. 1989) — аспирант, ассистент Российского химико-технологического университета им. Д. И. Менделеева

Плеханов Леонид Петрович (р. 1943) — кандидат технических наук, старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Полубелова Марина Игоревна (р. 1993) — студентка Санкт-Петербургского государственного университета

Попова Мария Сергеевна (р. 1994) — студентка Московского физико-технического института

Синицын Владимир Игоревич (р. 1968) — доктор физико-математических наук, доцент, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Стрижов Вадим Викторович (р. 1967) — доктор физико-математических наук, ведущий научный сотрудник Вычислительного центра им. А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук

Яковлев Олег Альбертович (р. 1992) — инженер-исследователь Орловского филиала Федерального исследовательского центра «Информатика и управление» Российской академии наук

Правила подготовки рукописей статей для публикации в журнале «Системы и средства информатики»

Журнал «Системы и средства информатики» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информационных технологий.

Журнал издается на русском языке. По специальному решению редколлегии отдельные статьи могут печататься на английском языке.

Тематика журнала охватывает следующие направления:

- информационно-телекоммуникационные системы и средства их построения;
- архитектура и программное обеспечение вычислительных машин, комплексов и сетей;
- методы и средства защиты информации.

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

Публикация не должна нарушать закон об авторских правах.

Направляя рукопись в редакцию, авторы сохраняют все права собственников данной рукописи и при этом передают учредителям и редколлегии неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на ее распространение в России и за рубежом. Каждый автор должен представить в редакцию подписанный с его стороны «Лицензионный договор о передаче неисключительных прав на использование произведения», формат которого размещен по адресу <http://www.ipiran.ru/publications/licence.doc>. Этот договор может быть представлен в бумажном виде (в 2-х экз.) или в электронном виде (отсканированная копия подписанного документа).

Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.

2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.
3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

Возвращение рукописи на доработку не означает ее принятия к печати.

Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам. Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.
5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.
6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или L^AT_EX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно.

7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.

Параметры страницы: формат — А4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху и снизу — 2, от края до нижнего колонтитула — 1,3.

Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

Рекомендуемый объем рукописи — не выше 20 страниц указанного формата.

Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

Все страницы рукописи нумеруются.

Шаблоны примеров оформления представлены в Интернете:

<http://www.ipiran.ru/publications/collected/template.doc>

8. Статья должна содержать следующую информацию на **русском и английском языках**:

- название статьи;
- Ф.И.О. авторов, на английском можно только имя и фамилию;
- место работы, с указанием города и страны и электронного адреса каждого автора;
- сведения об авторах, в соответствии с форматом, образцы которого представлены на страницах:
http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp и
http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
- аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных. Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком. В аннотации не должно быть ссылок на литературу и, по возможности, формул;
- ключевые слова — желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.
- источники финансирования работы (ссылки на гранты, проекты, поддерживающие организации и т. п.)

9. Требования к спискам литературы.

Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в каждом из списков литературы в порядке первых упоминаний.

Списки литературы представляются в двух вариантах:

- (1) **Список литературы к русскоязычной части.** Русские и английские работы — на языке и в алфавите оригинала.
- (2) **References.** Русские работы и работы на других языках — в латинской транслитерации с переводом на английский язык; английские работы и работы на других языках — на языке оригинала.

Необходимо для составления списка “References” пользоваться размещенной на сайте <http://www.translit.ru/> бесплатной программой транслитерации русского текста в латиницу, при этом в закладке «варианты. . . » следует выбрать опцию BNG.

Список литературы “References” приводится полностью отдельным блоком, повторяя все позиции из списка литературы к русскоязычной части, независимо от того, имеются или нет в нем иностранные источники. Если в списке литературы к русскоязычной части есть ссылки на иностранные публикации, набранные латиницей, они полностью повторяются в списке “References”.

Примеры ссылок на различные виды публикаций в списке “References”:

Описание статьи из журнала:

Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

Описание статьи из электронного журнала:

Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic exchange. *JCMC* 5(2). Available at: <http://www.ascusc.org/jcmc/vol5/issue2/> (accessed April 28, 2011).

Описание материалов конференций:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma “Novye resursosberegayushchie tekhnologii nedropol’zovaniya i povышeniya neftegazootdachi”* [6th Symposium (International) “New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact” Proceedings]. Moscow. 267–272.

Описание книги (монографии, сборники):

Lindorf, L. S., and L. G. Mamikonants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Описание переводной книги (в списке литературы к русскоязычной части необходимо указать: / Пер. с англ. — после названия книги, а в конце ссылки указать оригинал книги в круглых скобках):

1. В русскоязычной части:

Timoshenko С. П., Янг Д. Х., Уивер У. Колебания в инженерном деле / Пер. с англ. — М.: Машиностроение, 1985. 472 с. (Timoshenko S. P., Young D. H., Weaver W. Vibration problems in engineering. — 4th ed. — N.Y.: Wiley, 1974. 521 p.)

2. В англоязычной части:

Timoshenko, S. P., D. H. Young, and W. Weaver. 1974. *Vibration problems in engineering*. 4th ed. N.Y.: Wiley. 521 p.

Описание неопубликованного документа:

Latypov, A. R., M. M. Khasanov, and V. A. Baikov. 2004. Geology and production (NGT GiD). Certificate on official registration of the computer program No. 2004611198. (In Russian, unpubl.)

Описание интернет-ресурса:

Pravila tsitirovaniya istochnikov [Rules for the citing of sources]. Available at: <http://www.scribd.com/doc/1034528/> (accessed February 7, 2011).

Описание диссертации или автореферата диссертации:

Semenov, V. I. 2003. Matematicheskoe modelirovanie plazmy v sisteme kompaktnyy tor [Mathematical modeling of the plasma in the compact torus]. D.Sc. Diss. Moscow. 272 p.

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticeskogo slovarya informacionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

Описание ГОСТа:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. Moscow: Standardinform Publs. 10 p.

Описание патента:

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoj samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

10. Присланные в редакцию материалы авторам не возвращаются.
11. При отправке файлов по электронной почте просим придерживаться следующих правил:
 - указывать в поле subject (тема) название журнала и фамилию автора;
 - использовать attach (присоединение);
 - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
12. Журнал «Системы и средства информатики» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

Адрес редакции журнала «Системы и средства информатики»:

Москва 119333, ул. Вавилова, д. 44, корп. 2, ФИЦ ИУ РАН

Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05

e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич)

<http://www.ipiran.ru/journal/collected>

Requirements for manuscripts submitted to Journal “Systems and Means of Informatics”

Journal “Systems and Means of Informatics” publishes theoretical, review, and discussion articles on the research and development in the field of information technology.

The journal is published in Russian. By a special decision of the editorial board, some articles can be published in English.

Topics covered include the following areas:

- information and communication systems and tools of their design;
- architecture and software of computational complexes and networks; and
- methods and tools of information protection.

1. The Journal publishes original articles which have not been published before and are not intended for simultaneous publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editorial Board, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

Agreement on the transfer of rights to publish:

“We, the undersigned authors of the manuscript “. . .,” pass to the Founder and the Editorial Board of the Journal “Systems and Means of Informatics” the nonexclusive right to publish the manuscript of the article in Russian (or in English) in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.

Author(s) signature(s): (name(s), address(es), date).”

This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article has no classified data prohibited for publication.

2. A submitted article should be attached with **the data on the author(s)** (see item 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board and other authors about revisions and final approval of the proofs.
3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing, it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.
4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.
5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.

6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or L^AT_EX (.tex) formats, and additionally in the .pdf format. All documents may be sent by e-mail or provided on a CD or diskette. A hard copy submission is not necessary.

7. The recommended typesetting instructions for manuscript.

Pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer 1.3.

Text: font —Times New Roman, font size — 14, paragraph indent — 0.5, line spacing — 1.5, justified alignment.

The recommended manuscript size: not more than 20 pages of the specified format.

Use only standard abbreviations. Avoid abbreviations in the title and abstract. The full term for which an abbreviation stands should precede its first use in the text unless it is a standard unit of measurement.

All pages of the manuscript should be numbered.

The templates for the manuscript typesetting are presented on site:

<http://www.ipiran.ru/publication/collected/template.doc>

8. Articles should enclose data both in **Russian and English**:

- title;
- author's name and surname;
- affiliation — organization, its address with ZIP code, city, country, and official e-mail address;
- data on authors according to the format (see site):
http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp and
http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;

- abstract (not less than 100 words) both in Russian and in English. Abstract is a short summary of the article that can be published separately. The abstract is the main source of information on the article and it could be included in leading information systems and data bases. The abstract in English has to be an original text and should not be an exact translation of the Russian one. Good English is required. In abstracts, avoid references and formulae.

- Indexing is performed on the basis of keywords. The use of keywords from the internationally accepted thematic Thesauri is recommended.

Important! Keywords must not be sentences.

- Acknowledgments.

9. References. Russian references have to be presented both in English translation and in Latin transliteration (refer <http://www.translit.ru>, option BGN).

Please take into account the following examples of Russian references appearance:

Article in journal:

Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

Journal article in electronic format:

Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic

exchange. *JCMC* 5(2). Available at: <http://www.ascusc.org/jcmc/vol5/issue2/> (accessed April 28, 2011).

Conference proceedings:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povышeniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Books and other monographs:

Lindorf, L. S., and L. G. Mamikonants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Dissertation and Thesis:

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informacionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

State standards and patents:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. M.: Standardinform Publs. 10 p.

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoj samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

References in Latin transcription are presented in the original language.

References in the text are numbered according to the order of their first appearance; the number is placed in square brackets. All items from the reference list should be cited.

10. Manuscripts and additional materials are not returned to Authors by the Editorial Board.
11. Submissions of files by e-mail must include:
 - the journal title and author's name in the "Subject" field;
 - an article and additional materials have to be attached using the "attach" function;
 - an electronic version of the article should contain the file with the text and a separate file with figures.
12. "System and Means of Informatics" journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

Editorial Board address:

FRC CSC RAS, 44, block 2, Vavilov Str., Moscow 119333, Russia

Ph.: +7 (499)135 8692, Fax: +7 (495)930 45 05

e-mail: rust@ipiran.ru (to Prof. Rustem Seyful-Mulyukov)

http://www.ipiran.ru/english/journal_systems.asp

SYSTEMS AND MEANS OF INFORMATICS (СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ)

SCIENTIFIC JOURNAL

Volume 26 No.2 Year 2016

Editor-in-Chief and Chair of Editorial Council
Academician I. A. Sokolov

I N T H I S I S S U E:

SYSTEMS AND MEANS OF DEEP LEARNING FOR CLASSIFICATION PROBLEMS <i>O. Yu. Bakhteev, M. S. Popova, and V. V. Strijov</i>	4
SELF-TIMING ANALYSIS OF ELECTRONIC CIRCUITS ON THE LOWER LEVEL OF HIERARCHY <i>L. P. Plekhanov</i>	23
LEXICAL ANALYSIS OF DYNAMICALLY GENERATED STRING EXPRESSIONS <i>M. I. Polubelova and S. V. Grigorev</i>	43
NORMAL PUGACHEV FILTERS FOR STATE LINEAR AUTOCORRELATED STOCHASTIC SYSTEMS <i>I. N. Sinitsyn and E. R. Korepanov</i>	63
MODIFIED ELLIPOIDAL SUBOPTIMAL FILTERS FOR NONLINEAR STOCHASTIC SYSTEMS ON MANIFOLDS <i>I. N. Sinitsyn, V. I. Sinitsyn, and E. R. Korepanov</i>	79
GENERATING REALISTIC STRUCTURE-FROM-MOTION DATASETS THROUGH VIRTUAL PHOTOGRAPHY <i>O. A. Yakovlev and A. V. Gasilov</i>	98
HIERARCHICAL FORMAT OF A BIOGRAPHICAL FACT <i>I. M. Adamovich and O. I. Volkov</i>	108
INFORMATION SYSTEM OF CELLULAR AND NEURAL NETWORK MODELING OF CHEMICAL ACCIDENT CONSEQUENCES AT DANGEROUS INDUSTRIAL FACILITIES <i>S. P. Dudarov and P. L. Papaev</i>	123