

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(167)
2010

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

КОМПЬЮТЕРНАЯ ГРАФИКА

- Кривцов О. А., Кориков А. М. Методы отслеживания головы человека на видео-кадрах: состояние проблемы 2
Вахромцев Е. М. Применение методов "живого" моделирования для производственного мониторинга. 10

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Жук Д. М., Маничев В. Б., Ильницкий А. О. Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временной области. Часть 1 16
Перепелкин Е. А., Поддубный В. И., Варкентин А., Ган М. Применение прикладного пакета САМеL-View для моделирования управляемого движения колесного трактора. 24

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Косников Ю. Н. Геометрическое моделирование и отображение местности по топографическим данным 30
Немтинов В. А., Горелов А. А., Немтинов К. В., Кулин Д. В. Пространственно-временная модель объекта культурно-исторического наследия. 36
Богданов М. Р., Богданова А. В., Габидуллин Ю. З. Беспроводные сенсорные сети в экологии 39

НЕЙРОИНФОРМАТИКА

- Осипов В. Ю. Рекуррентная нейронная сеть с управляемыми синапсами. 43
Крючин О. В. Нейросетевой симулятор с автоматическим подбором активационных функций 47

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

- Жуков О. Д. Информационные технологии на основе полиномиальной алгебры 52
Калитин Д. В. Проектирование параллельных декомпозиций управляющих автоматов в k -значных логиках 60

ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Михеева Т. И., Сапрыкин О. Н. Паттерны проектирования подсистемы анализа состояния пространственно-координированных объектов интеллектуальной транспортной системы 64
Чичко А. Н., Сачек О. А., Лихозов С. Г. Методы автоматизации обработки изображений микроструктур перлитных сталей 71

ОБМЕН ОПЫТОМ

- Кузнецов А. В., Суриков Е. Н. Проблемы внедрения ERP-систем при разработке и мелкосерийном производстве высокотехнологичной продукции 77

Contents

Приложение. Я. Е. Львович, С. Ю. Белецкая. Адаптивные методы оптимального проектирования

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НАРИНЬЯНИ А. С.

НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.932.2

О. А. Кривцов, аспирант,
А. М. Корилов, д-р техн. наук,
проф., зав. кафедрой,

Томский государственный университет систем
управления и радиоэлектроники,
e-mail: olegkrivtsov@mail.ru, korikov@asu.tusur.ru

Методы отслеживания головы человека на видеокадрах: состояние проблемы

Рассмотрены области применения методов отслеживания положения головы человека на видеокадрах. Проведен ретроспективный анализ методов отслеживания головы человека и отмечены факторы, усложняющие применение известных методов.

Ключевые слова: трекинг головы, видеокадры, методы трекинга, развитие, сложности трекинга

Введение

Видеоизображение головы человека, а особенно его лица, содержит чрезвычайно важную информацию о человеке, сферах его бытия и деятельности. В связи с этим изображение лица (голова) человека является интересным объектом анализа на видеокадрах.

В настоящее время проблема отслеживания головы на основе анализа видеокадров не решена в объеме, позволяющем устойчиво отслеживать голову человека на видеокадрах в течение продолжительного времени. Это обусловлено многими факторами, усложняющими анализ изображения головы человека, такими как неравномерность освещения сцены, изменение формы и текстуры головы за счет изменения мимики лица и т. д. Актуальность подтверждается множеством научных трудов, призванных приблизить решение этой проблемы и опубликованных за последние тридцать лет.

В данной статье сделана попытка систематизации результатов исследований по решению проблемы отслеживания головы человека на видеокадрах: представлены основные области применения; изложена история развития методов; перечислены основные трудности, встающие перед исследователем при разработке алгоритмов и программных средств для решения задачи отслеживания головы.

Применение методов отслеживания головы человека на видеокадрах

Отслеживание объекта на видеокадрах — это решение задачи последовательного определения положения объекта, независимо от того, движется объект или камера [1]. Целью отслеживания головы человека (англ. *head tracking*, трекинг головы) является последовательное определение положения головы по ее изображению на видеопоследовательности [2]. Рассмотрим кратко области, в которых возникают подобные задачи.

Телекоммуникации и телеконференции в настоящее время становятся распространенными в сети Интернет. Чтобы реализовать автоматическое распознавание событий, связанных с формальными и неформальными встречами людей, необходимо понимать, где находятся участники беседы и извлекать информацию об их поведении [3]. При анализе визуальных сигналов используются алгоритмы компьютерного зрения, в том числе трекинга головы. Кодирование движений головы и выражения лица осуществляется по известным стандартам, таким как MPEG-4 (<http://www.chiariglione.org/mpeg/>), и позволяет получать для телекоммуникаций и телеконференций потоки данных, нетребовательные к пропускной способности сети [4].

В компьютерной графике трекинг может применяться для управления аватаром [5]. Аватар — анимированное изображение, используемое для персонализации пользователя и обычно не являющееся его настоящей фотографией. В качестве примера можно привести приложение Logitech Video Effects компании Logitech или приложение ManyCam (<http://manycam.com>) компании CamShare, в которых пользователь управляет своим аватаром движением головы.

Трекинг головы человека также может быть использован для построения трехмерных моделей путем склейки фрагментов изображений, полученных от стереокамеры [6]. Трекинг положения головы играет роль первого шага в процессе построения трехмерной модели путем регистрации (сопоставления) и определения положения отснятых частей модели друг относительно друга.

В системах видеонаблюдения трекинг головы человека может применяться для определения подозрительных действий или нежелательных событий [7, 8].

Отслеживание особенных "характерных" точек (англ. *feature points*) лица человека играет существенную роль при создании *алгоритмов распознавания речи* путем анализа движений губ [4].

В устройствах видео- и фотосъемки трекинг находит применение в фокусировке и стабилизации устройств съемки [9], таких как фотокамера. Цифровая стабилизация изображения — это технология, применяемая в видеосъемочной аппаратуре, позволяющая (помимо компенсации движения камеры) полностью или частично компенсировать движение одного из объектов в кадре и улучшать качество изображения. Автоматическая фокусировка — это технология, позволяющая удерживать объект в поле зрения камеры при их взаимном перемещении. Например, в веб-камере Logitech QuickCam Fusion (<http://logitech.com/>) реализована возможность автоматического наведения фокуса на лицо человека и подстройки фокуса при перемещении головы.

В робототехнике отслеживание головы человека применяется для обеспечения взаимодействия робота с окружающими людьми [1, 10], при создании ассистентов роботов и их автономной навигации [11].

В системах поиска информации трекинг головы применяется для выборки из базы мультимедиа данных [7]. Видеоданные сложно автоматически индексировать по содержанию, и трекинг может способствовать тому, что компьютер самостоятельно примет решение о характере действий человека на видеокдрах.

В медицине трекинг мог бы использоваться для получения панорамных изображений [12], для психологического анализа выражения лица [13].

"Направление головы человека говорит многое о его намерениях, что может использоваться для анализа действий человека. Если рабочий робот или инвалидная коляска сможет управляться движениями головы, это будет способствовать снижению стресса пациента" [14].

В человеко-машинном взаимодействии одной из наиболее перспективных областей применения трекинга головы является создание новых способов ввода информации в компьютер. Перцептуальные (от англ. *perception*, восприятие) пользовательские интерфейсы обеспечат компьютер аналогами человеческих чувств, например, воспроизведением звуков и речи, восприятием прикосновений, а в нашем случае — способностью видеть [15].

Программы, в которых управление основано на компьютерном зрении, пока встречаются редко, несмотря на возможности: "хэндс фри" управление курсором мыши, управление в компьютерных играх и навигация в трехмерных мирах [16]. "В настоящее время с ростом уровня использования компьютеров во всех сферах человеческой

деятельности все более остро встает задача построения как можно более удобных интерфейсов человек—компьютер. Одним из основных направлений в этой области является создание естественного человеко-машинного интерфейса, позволяющего пользователю вводить информацию в компьютер наиболее привычными и естественными для него способами: с помощью речи, жестов, мимики и т. п." [17].

Многие исследователи предложили управлять курсором и выдавать клики мышью путем движений головы, глаз, лица. Такие системы требуют надежной и быстрой стратегии трекинга лица. Некоторые из существующих решений еще не достигли необходимого уровня устойчивости и точности [18].

Одним из подходов является распознавание направления взгляда пользователя для указания положения курсора на экране. Однако согласно ряду исследований существенная часть движений глаз совершается человеком неосознанно, что делает привязку курсора к направлению взгляда нецелесообразным. Другой подход основан на том, что, в отличие от глаз, движения головы лучше поддаются сознательному контролю со стороны пользователя, что позволяет опираться на них при управлении компьютером [17]. В качестве примера такой системы приведем программу hMouse [18], которая использует отслеживание углов наклона головы для позиционирования курсора на экране.

Портативные компьютеры, например ноутбуки, часто бывают недостаточно удобны для позиционирования курсора мыши. Управление программами на основе трекинга могло бы быть заменой неудобному трекболу, который сейчас используется во многих ноутбуках.

Среди пользователей компьютера есть люди с нарушением опорно-двигательного аппарата, пожилые люди, неспособные справиться с клавиатурой и мышью [17, 18]. Перцептуальные интерфейсы могут помочь таким пользователям движением головы управлять курсором [16].

С удешевлением LCD-мониторов появляется все большее число пользователей, имеющих несколько мониторов на одном компьютере. Это доставляет определенные неудобства в работе, так как требует приложения больших усилий для перемещения курсора мыши, перетаскивания объектов с одного монитора на другой. Актуальным представляется автоматическое отслеживание головы пользователя и перемещение курсора мыши с одного монитора на другой после определения направления взгляда пользователя, как, например, это реализовано в программе MagicPointing [19].

В интерактивных средах, таких как автоматизированные точки продажи или приборная панель управления самолетом, отслеживание на-

правления взгляда может использоваться для прямого указания, когда руки и/или ноги реализуют другое действие, либо как дополнительная информация, когда действие требует многих параметров ввода [6].

Для стабилизации изображения лица человека могут применяться, например, методы трекинга головы, основанные на трехмерных геометрических моделях головы. Такие методы теоретически способны генерировать стабилизированные изображения лица, инвариантные к поворотам головы [20], что может способствовать более устойчивой работе алгоритмов анализа эмоций человека [21] и идентификации человека по изображению лица [22].

Методы распознавания эмоций могут классифицировать изображения лица человека как принадлежащие к одной из категорий обычных эмоций, таких как радость, грусть, злость и т. д. Некоторые из них способны выражать жесты лица в виде единиц действия системы FACS [23] — это описание движений мускулов лица и языка, основанного на анализе анатомического строения лица.

К ограничениям практически всех методов автоматического распознавания эмоций относится то, что они предназначены для специальных условий съемки, которые устраняют другие факторы, усложняющие анализ. Автоматическое распознавание эмоций лица в динамической обстановке является сложной проблемой, включающей в себя "жесткое" движение головы (движение головы в целом как твердого тела), нефронтальное положение головы по отношению к камере, перекрывание частей лица из-за движения головы, ношения очков, разговора, невыразительной мимики лица [24].

Приведем в качестве примера задачу анализа мимики лица водителя автомобиля для оценки его эмоционального состояния и ассистирования (создание интеллектуального транспортного средства) [25]. Нежесткое движение головы водителя (движения рта, глаз—бровей, щек) может сказать многое о его ментальном состоянии, например, является ли водитель агрессивным, усталым или расслабленным.

Современные системы идентификации человека по изображению его лица в базе данных [22] все еще зависят от ориентации лица по отношению к камере. Например, система Visionics может распознавать людей при углах наклона не более 20° [26]. Полученное на выходе трекера головы "стабилизированное" фронтальное изображение лица может помочь в улучшении такой системы. Также трекинг головы можно применить для определения видеокadra, на котором лицо выглядит наиболее фронтально, а затем этот кадр подавать на вход системы идентификации лица [27].

Трекинг головы может использоваться для создания входного модуля для системы анимации лица [26]. В настоящее время интерес к исследованиям в области анимации лица подогревается появлением огромного числа виртуальных персонажей в фильмах [28]. При создании компьютерных эффектов для художественных фильмов применяется методика, позволяющая оживлять анимационные персонажи путем движения людей-актеров в реальном масштабе времени. Эта методика имеет название (дословно) "анимация, движимая действиями", *action-driven animation*. В данном подходе на некоторые части тела актера (лицо, руки) прикрепляются датчики, с помощью которых определяется и отслеживается положение в пространстве неких контрольных точек [13]. Процесс создания анимации лица требует интенсивного вмешательства человека и кропотливой настройки параметров. Актуальной проблемой в данной области является внедрение неинвазивных методов отслеживания головы актера, т. е. без прикрепления датчиков, а лишь путем анализа видеокadров, поступающих с видеокамеры.

Развитие методов отслеживания головы человека на видеокadрах

Важным событием в истории развития методов визуального трекинга стало создание в 1981 г. Дж. Лукасом и Т. Канаде алгоритма сопоставления (или регистрации) двух изображений [29], который по классификации, предложенной С. Бэкером и И. Мэтьюсом [30] также можно назвать прямым аддитивным алгоритмом (англ. *forwards-additive algorithm*). В данной методике используется пространственный градиент интенсивности пикселей изображения, а параметры наилучшего совпадения изображений определяются алгоритмом Ньютона—Рафсона. На каждой итерации этого алгоритма первое из изображений деформируется с использованием параметров, найденных на предыдущей итерации, и вычисляется значение оптимизационного критерия как взвешенная сумма квадратов разностей соответствующих пикселей изображений. Затем параметры обновляются с помощью метода градиентного поиска.

Метод [29] на момент его создания являлся наиболее быстродействующим, что объяснялось анализом гораздо меньшего числа потенциальных вариантов, чем в существовавших на тот момент аналогах. Лукас и Канаде показали, каким образом можно обобщить этот алгоритм на случаи перемещения, вращения, масштабирования и других моделей движения. Алгоритм [29] стал фактическим стандартом для использования во многих задачах компьютерного зрения, в том числе и в задаче отслеживания лица и головы человека [31].

Варианты этого алгоритма используются практически во всех схемах сжатия изображений с компенсацией движения, таких как MPEG или H.263 [32]. В последующие годы появились многочисленные модификации алгоритма [30, 33—35] и аналогичные ему методы [12, 36]. Основные направления разработок были связаны с улучшением устойчивости [30] и вычислительной производительности алгоритма, например, за счет улучшения скорости деформации изображений. В качестве интересного примера можно привести попытку реализовать итеративный процесс оптимизации с помощью генетического алгоритма [35].

Ранние подходы к отслеживанию головы человека упрощенно предполагали, что голова не может испытывать сильных отклонений вне плоскости видеокadra, поэтому их можно назвать двумерными (2D) методами. Такие методы опирались на задание и отслеживание характерных точек лица [37, 38], устойчивых градиентных краев, контуров [39] или на информации о цвете изображения [40]. Это объяснялось тем, что они были вычислительно эффективны и сравнительно просты в реализации, что было критическим фактором при тогдашнем уровне развития вычислительной техники.

Одним из подходов являлось детектирование устойчивых градиентных краев (*edges*). Отслеживание краев является естественно стабильным к изменениям освещения, даже для освещающих материалов, что не всегда верно для методов, использующих внутренние пиксели проекции объекта [1].

Информация о статистическом распределении цветовой информации изображения также нашла применение в алгоритмах трекинга лиц. Так, в 1998 г. Г. Брадски создал алгоритм CAMSHIFT [15], который на основе цветовой информации был способен отслеживать лица с четырьмя степенями свободы движения.

Направление, основанное на построении геометрических текстурных моделей головы человека, также постепенно эволюционировало [41] от двумерных подходов (например, двумерный эллипс [40], прямоугольная плоская модель [12]) до полностью трехмерных методов (например, эллипсоидальная модель головы [3, 4, 42]).

Приемлемую точность показала цилиндрическая модель головы [20], впервые предложенная М. Ла Каскиа (M. La Cascia) и другими исследователями в 1998 г. Развивает идею работа [43], где моделируется изменение условий освещения. Заслугой этих исследователей также является то, что они подготовили базу данных видеоизображений (<http://www.cs.bu.edu/groups/ivc/HeadTracking>), которые можно применять в экспериментальных исследованиях по отслеживанию головы человека.

По мнению авторов, наиболее распространенной моделью головы в различных работах является трехмерная цилиндрическая модель головы, что, на наш взгляд, объясняется сравнительной легкостью данного подхода в реализации и обеспечиваемой им хорошей точности. Идею использования в качестве модели головы цилиндра, покрытого текстурой, развивают работы [21, 26, 31, 44, 45].

Разработка методов реалистичной анимации лица [28] человека берет свое начало еще с основополагающей работы Ф. Парка [46] в 1972 г. С развитием ЭВМ осуществляются попытки создания модели лица, пригодной для анимации. К таким моделям относится трехмерная сеточная модель *Candide*, которая была создана М. Ридфолком в 1987 г. [47]. Эта работа была вдохновлена попыткой кодировать видеоизображение лица путем отслеживания лица на видеокadre и его анимации [48]. Впоследствии модель *Candide* несколько раз модифицировалась в целях достижения более реалистичного результата. Последняя версия этой модели называется *Candide-3* [49]. Она состоит из трехмерной проволочной сетки из 113 вершин и 184 треугольников. Модель *Candide* и ее модификации широко распространены из-за своей простоты.

В 1998 г. Дж. Эдвардсом предложена деформируемая модель лица [50] под названием *Active Appearance Model* (AAM), позволяющая успешно отслеживать мимику лица пользователя и ее анимации. Например, модель AAM успешно используется в работе [25] для анализа мимики лица водителя автомобиля. Развитию методов трекинга, основанных на этой модели послужили работы [51—54].

Развивается также и направление трекинга лица и головы человека с использованием методов распознавания образов. Ранние подходы использовали метод главных компонент (PCA) для создания моделей визуальных объектов, основанных на собственных пространствах (*eigenspaces*) [55, 56] или метод опорных векторов (SVM) [57, 58].

Использовались и различные топологии нейронных сетей, такие как сети вейвлетов Габора [59—62] или ассоциативная память [63]. Сети вейвлетов — многослойные нейронные сети, в которых вейвлеты играют роль активационных функций, впервые были предложены Зангом и Бенвенистом в 1992 г. [64].

Развитие группы методов трекинга головы человека, основанных на распознавании образов, базируется на алгоритме [65], разработанном в 2000 г. П. Виолой и М. Джонсом. Этот алгоритм основан на представлении изображения в специальном "интегральном виде" и построении каскада из "слабых" классификаторов, так называемых характеристик Хаара (англ. *Haar features*). Эта сис-

тема предлагала значительный прирост производительности по отношению к лучшим аналогам, что позволило создать реализацию, способную обрабатывать видеокadres в реальном времени на среднем компьютере и даже на мобильных устройствах. Возможна тренировка каскада на распознавание объекта с незначительной вариацией углов обзора, однако этот алгоритм в большой степени является двумерным алгоритмом.

Предложенный Виолой и Джонсом детектор лица [65] стимулировал большой интерес к разработке более общих систем [66] и подтолкнул разработку детекторов головы независимо от направления съемки (мультивидовых детекторов, *multi-view face detectors* [67—70]). В алгоритмах этого направления классификаторы формируются в древовидную структуру таким образом, чтобы одновременно проводить классификацию (объект/не объект) и определять ориентацию объекта в пространстве.

На практике бывает сложно реализовать устойчиво работающий алгоритм отслеживания головы, использующий только один подход к этой проблеме. Часто алгоритмы комплексуются и объединяются в один более устойчивый алгоритм [58].

Так, недостатком методов трекинга с помощью геометрических моделей головы является их неспособность инициализировать себя; для их использования нужно задать первоначальное положение. В ранних работах инициализация системы проводилась в основном вручную, когда пользователя просили поместить лицо точно в кадр и посмотреть в камеру либо указать область кадра, в которой находится лицо. С развитием алгоритмов распознавания образов для инициализации моделей головы начинают применяться алгоритмы детектирования лица на видеокadre. Например, в системе hMouse [18] детектирование применяется только для инициализации модели головы, а затем проводится отслеживание головы с помощью метода, приведенного в работе [15].

Вместе с тем, алгоритмы детектирования лица (голова) обладают сравнительно низкой точностью отслеживания, тогда как методы, использующие геометрические модели, обеспечивают высокую точность. Комбинирование данных от различных алгоритмов на основе методов математической статистики, таких как рекурсивный оптимальный фильтр Калмана или *particle filter* (последовательный метод Монте-Карло), часто позволяет добиться как более высокой точности, так и лучшей устойчивости метода.

С момента появления первых прототипов веб-камер в 1991 г. прошло не так уж много времени, но они уже превратились во вполне сформировавшийся, отдельный класс устройств. В настоящее время распространены как дешевые камеры, под-

ключаемые через интерфейс USB и устанавливаемые на мониторе, так и веб-камеры, встраиваемые в мониторы и ноутбуки. Это подтолкнуло разработку алгоритмов человеко-машинного взаимодействия, позволяющих управлять компьютером путем анализа видеоизображения пользователя. Так, активно исследуется проблема управления курсором мыши путем отслеживания носа пользователя [16], либо его головы [17, 18]. М. Эшдаун [19] обратил внимание на сложности в управлении приложениями при наличии нескольких мониторов и предложил использовать трекинг головы человека для переключения между мониторами.

Факторы, усложняющие трекинг головы человека

Отслеживание головы человека является сложной, нетривиальной задачей. Кратко рассмотрим факторы, усложняющие решение данной задачи.

Качество видео. Оно существенно зависит от используемой веб-камеры. В настоящее время популярны дешевые веб-камеры, которые обеспечивают видеоизображение сравнительно низкого разрешения и низкого качества, с наличием шума, волнообразного изменения освещенности, наличием квантизации [2]. Неравномерная или низкая частота кадров, поступающих с камеры, также является фактором, усложняющим отслеживание головы (изображение головы может испытывать значительные перемещения между кадрами).

Потеря информации при проецировании 3D-объекта на 2D-изображение. Так как голова человека является объемным трехмерным телом сложной формы, при проецировании ее на плоскость видеокadre часть информации неизбежно оказывается утерянной [11]. Например, при повороте головы нос может скрыть часть щеки (англ. термин — самоперекрытие, *self occlusion*). По этой причине в алгоритмах, основанных на построении модели головы учитывают видимость частей лица и головы в данный момент времени (например, для этих целей используется буфер глубины в библиотеке компьютерной графики OpenGL (<http://opengl.org>)). Также вводят специальное изображение-карту уверенности (англ. *confidence map*), интенсивность (цвет) фрагментов которой указывает степень потери информации для каждого спроецированного фрагмента головы [31].

Сложность формы и движения объекта. Твердое (недеформируемое) тело способно перемещаться в трехмерном пространстве с шестью степенями свободы движения и может поворачиваться как в плоскости изображения, так и вне нее. Также возможно, что камера перемещается относительно головы, т. е. меняется точка наблюдения [71].

В общем же случае голова является деформируемым телом. Деформации возникают за счет из-

менения мимики лица, движений челюсти. Поэтому существуют два вида моделей головы: жесткие [31, 42, 43] и деформируемые [72—74]. Соответственно и движение разделяют на жесткое (англ. *rigid*), связанное с перемещением и вращением в пространстве тела целиком, и нежесткое (англ. *non-rigid*), связанное со смещением деформируемых частей лица. Сложность формы головы приводит к усложнению ее моделей, а иногда и персонализации модели, т. е. необходимости ее подгонки под определенного пользователя.

Выбросы. Алгоритмы компьютерного зрения и, в частности, алгоритмы трекинга головы чувствительны к ошибкам. Когда эти ошибки характеризуются хорошо известным распределением, можно применить методы математической статистики для их компенсации. Однако иногда возникают выбросы (англ. *outliers*). Выбросы могут возникать вследствие неверных предположений, числовых ошибок, недостатков эвристических закономерностей. Известно, что даже небольшое число выбросов может существенно ухудшить результат алгоритма [75].

Изменение освещенности. Освещенность сцены и объекта нестационарна во времени и в пространстве. При наличии одного источника освещения голова неминуемо будет освещена лучше с одной стороны и хуже с другой. Поэтому в ряде подходов к трекингу головы вводятся дополнительные методики для учета освещенности [12]. Например, в [31] используют цилиндрическую модель головы, покрытую текстурой, и при определении положения головы на каждом кадре осуществляют обновление текстуры, что делает метод более устойчивым к изменениям освещения, так как изменениями освещенности между двумя последовательными кадрами можно пренебречь. Еще одной возможной мерой является ввод условий освещенности в выражение для поиска параметров перемещения модели головы [43].

Требования к производительности метода. Очевидно, что учет всего многообразия условий съемки может привести к значительному росту необходимых вычислительных ресурсов компьютера [7]. Многие же задачи требуют работы в реальном масштабе времени (не менее 10 кадров/с) и умеренного потребления процессорного времени (около 20—30 %), так как компьютер должен выполнять и другие задачи.

Итак, для успешного отслеживания головы человека существует много препятствий. На практике пока не создан общий метод, который бы успешно преодолевал все из перечисленных факторов. Поэтому исследователям приходится делать определенные предположения о характере допущений, при которых предлагаемый метод спосо-

бен работать устойчиво. Отметим, что устойчивость — необходимое условие работоспособности любого метода (системы). Однако для обозначенных выше областей применения важно дополнительное свойство робастности. Говорят, что метод отслеживания головы человека является робастным, если он способен работать заданный (требуемый) период времени без сбоев, справляясь с возможными вариациями, иногда непредсказуемыми, внешних условий с минимальной потерей функциональности (<http://en.wikipedia.org/wiki/Robust>), т. е. оценивая параметры перемещения головы человека с заданной точностью в заданный период времени.

Заключение

Отслеживание лица и головы человека на видеокдрах привлекает внимание многих исследователей в течение последних тридцати лет. Однако проблема устойчивого отслеживания головы человека остается актуальной из-за наличия серьезных трудностей, возникающих при трекинге головы: вариации качества видео, сложность формы головы и ее движения, неравномерность освещения и т. п.

Отслеживание головы человека на видеокдрах, полученных в обычных условиях съемки, применяется или обладает потенциалом применения для создания систем трехмерной компьютерной анимации, систем видеонаблюдения (визуальный анализ действий объекта наблюдения), интерфейсов взаимодействия человек—компьютер, распознавания эмоций человека и стабилизации изображения лица человека в кадре, медицинской диагностики и лечения, видеоконференций.

Проведенный ретроспективный анализ проблемы позволил выделить основные этапы развития методов отслеживания головы человека и основные классы методов, основанных на: геометрических моделях головы, цветовой статистике, детектировании и отслеживании характерных точек.

Список литературы

1. Lepetit V., Fua P. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey // Foundations and Trends in Comp. Graphics and Vision. 2005. Vol. 1. N 1. P. 1—89.
2. Gangaputra S., Geman D. A Unified Stochastic Model for Detecting and Tracking Faces // Comp. and Robot Vision, Canadian Conf. Los Alamitos, CA, USA: IEEE Computer Society. 2005. P. 306—313.
3. Xiong Y., Quek F. Head Tracking with 3D Texture Map Model in Planning Meeting Analysis // Proc. Intl. Workshop on Multimodal Multiparty Meeting Processing, Trento, Italy, 2005. P. 1—8.
4. Bottino A. Real Time Head and Facial Features Tracking from Uncalibrated Monocular Views // Proc. 5th Asian Conf. on Comp. Vision. Melbourne, Australia: Asian Federation of Comp. Vision Societies. 2002. Vol. 2. P. 776—781.

5. **Ström J.** Model-Based Real-Time Head Tracking // EURASIP Journal on Applied Signal Processing. New York, NY, USA: Hindawi Publishing Corp. 2002. Vol. 2002. N 1. P. 1039–1052.
6. **Morency L.** Stereo Based Head Pose Tracking Using Interactive Closest Point and Normal Flow Constraint // AI Tech. Report 2003-006. 2003. 60 P.
7. **Yilmaz A., Javed O., Shah M.** Object Tracking: A Survey ACM Computing Surveys. New York, NY, USA: ACM. 2006. Vol. 38. N 4. Article N 13.
8. **Avidan S.** Ensemble Tracking // IEEE Tran. Pattern Analysis and Machine Intelligence. New-York, NY, USA: IEEE Computer Society. 2007. Vol. 29. N 2. P. 261–271.
9. **Li Y., Ai H., Huang C., Lao S.** Robust Head Tracking Based on a Multi State Particle Filter // Proc. 7th Intl. Conf. on Automatic Face and Gesture Recognition. Southampton, UK: IEEE Computer Society, 2006. P. 335–340.
10. **Malis E.** Survey of vision-based robot control // European Naval Ship Design, Captain Comp. IV Forum, ENSIETA, Brest, France, 2002. 16 p.
11. **Ablavsky V., Thangaly A., Sclaroff S.** Layered Graphical Models for Tracking Partially Occluded Objects // Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Anchorage, AK. 2008. P. 1–8.
12. **Hager G., Belhumeur P.** Efficient Region Tracking With Parametric Models of Geometry and Illumination // IEEE Tran. Pattern Analysis and Machine Intelligence. Washington, DC, USA: IEEE Computer Society. 1998. Vol. 20. P. 1025–1039.
13. **Wei X., Zhu Z., Yin L., Ji Q.** A Real Time Face Tracking and Animation System // Comp. Vision and Pattern Recognition Workshop. Los Alamitos, CA, USA: IEEE Computer Society. 2004. Vol. 5. P. 71.
14. **Ishii S., Arita D., Taniguchi R.** Real-Time Head Pose Estimation with Stereo Vision // Proc. 9th Korea-Japan Joint Workshop on Frontiers of Comp. Vision. 2003. P. 79–83.
15. **Bradski G.** Computer Vision Face Tracking for Use in Perceptual User Interface // Intel Technology Journal. Intel, 1998. Vol. 2. P. 15.
16. **Gorodnichy D., Malik S., Roth G.** Mouse "Use Your Nose as Mouse" — A New Technology for Hands Free Games and Interfaces // Image and Vision Computing. 2004. Vol. 22. N 12. P. 931–942.
17. **Вежневцев В., Шоргин Р., Вежневцев А.** Система управления курсором "мышь" при помощи движений головы пользователя // Труды конф. по компьютер. графике и машинному зрению Graphicon-2006. Новосибирск, 2006. С. 455–458.
18. **Fu Y., Huang T.** hMouse: Head Tracking Driven Virtual Computer Mouse // IEEE Workshop on Applications of Comp. Vision. Washington, DC, USA: IEEE Computer Society. 2007. P. 30.
19. **Ashdown M., Oka K., Sato Y.** Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors // Proc. of Conf. on Human Factors in Computing Systems, Portland, OR, USA. 2005. P. 1188–1191.
20. **Cascia M., Isidoro J., Sclaroff S.** Head Tracking via Robust Registration in Texture Map Images // In Proc. IEEE Conf. on Comp. Vision and Pattern Recognition. Los Alamitos, CA, USA : IEEE Computer Society. 1998. P. 508–514.
21. **Kumano S., Otsuka K., Yamato J., Maeda E., Sato Y.** Pose-Invariant Facial Expression Recognition Using Variable Intensity Templates // Proc. Asian Conf. on Comp. Vision. Part I, LNCS 4843. 2007. P. 424–434.
22. **Wiskott L., Fellous J. M., Kruger N., Malsburg C.** Face Recognition by Elastic Bunch Graph Matching // IEEE Trans. Pattern Anal. Mach. Intell. Washington, DC, USA: IEEE Computer Society. 1997. Vol. 19. N 7 P. 775–779.
23. **Cascia M., Valenti L., Sclaroff S.** Fully Automatic, Real-Time Detection of Facial Gestures from Generic Video // Proc. IEEE 6th Workshop on Multimedia Signal Processing. Siena, Italy. 2004. P. 175–178.
24. **Moriyama T., Kanade T., Cohn J. F., Xiao J., Ambadar Z., Gao J., Imamura H.** Automatic Recognition of Eye Blinking in Spontaneously Occurring Behavior // Proc. 16th Intl. Conf. on Pattern Recognition. Washington, DC, USA: IEEE Computer Society. 2002. Vol. 4. P. 78–81.
25. **Baker S., Matthews I., Xiao J., Gross R., Kanade T.** Real-time Non-rigid Driver Head Tracking For Driver Mental State Estimation // Tech. Report CMU-RI-TR-04. Pittsburgh, PA, USA: Robotics Institute. 2004. 12 p.
26. **Brown L.** 3D Head Tracking Using Motion Adaptive Texture-Mapping. IEEE Comp. Society Conference on Comp. Vision and Pattern Recognition. 2001. Vol. 1. P. 998.
27. **Yang Z., Ai H., Wu B., Lao S., Cai L.** Face Pose Estimation and its Application in Video Shot Selection // Proc. of 17th Intl. Conference on Pattern Recognition. IEEE Computer Society. 2004. Vol. 1. P. 322–325.
28. **Deng Z., Newman U.** Data Driven 3D Facial Animation Springer. 2008. 296 p.
29. **Lucas B., Kanade T.** An Iterative Image Registration Technique with an Application to Stereo Vision // Proc. 7th Intl. Joint Conf. on Artificial Intelligence. Vancouver, British Columbia. 1981. P. 674–679.
30. **Baker S., Matthews I.** Lucas-Kanade 20 Years On: A Unifying Framework: Part 1 // Tech. report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University. Pittsburg, PA, USA. 2002. 47 p.
31. **Xiao J., Moriyama T., Kanade T., Cohn J.** Robust Full-Motion Recovery of Head by Dynamic Templates and Registration Techniques // Intl. Journal of Imaging Systems and Technology. New York, NY, USA: Wiley. 2003. Vol. 13. P. 85–94.
32. **Szeliski R.** Image Alignment and Stitching: A tutorial // Foundations and Trends in Comp. Graphics and Comp. Vision. Hanover, MA, USA: Now Publishers Inc. 2006. Vol. 2. N 1. P. 1–104.
33. **Rav-Acha A., Peleg S.** Lucas-Kanade without Iterative Warping // Proc. IEEE Intl. Conf. on Image Processing, 2006. Atlanta, GA, USA. P. 1097–1100.
34. **Junghans M., Leich A., Jentschel H.** Lucas-Kanade Algorithm with GNC // Proc. 7th Intl. Conf. on Signal Processing, 2004. Beijing, China, 2004. Vol. 2. P. 1088–1091.
35. **Tagliassachi M.** A Genetic Algorithm for Optical Flow Estimation // Image and Vision Computing. Newton, MA, USA: Butterworth-Heinemann. 2007. Vol. 25. N 7. P. 141–147.
36. **Sepp W., Hirzinger G.** Real-Time Texture-Based 3D Tracking // Pattern Recognition. Berlin, Germany: Springer Berlin / Heidelberg, 2003. Vol. 2781/2003. P. 330–337.
37. **Colmenarez A., Lopez R., Huang T.** 3D Model-Based Head Tracking // Proc. of Intl. Conf. VSIP. SPIE Press 1997. P. 426–434.
38. **Yao P., Evans G., Calway A.** Face Tracking and Pose Estimation Using A-ne Motion Parameters // Proc. 12th Scandinavian Conf. on Image Analysis. Bergen, Norway, 2001. P. 531–536.
39. **Isard M., Blake A.** Condensation. Conditional Density Propagation for Visual Tracking // International Journal on Comp. Vision. Netherlands: Kluwer Academic Publishers. 1998. Vol. 29 (1). P. 5–28.
40. **Birchfield S.** Elliptical Head Tracking Using Intensity Gradients and Color Histograms // Proc. IEEE Conf. on Comp. Vision and Pattern Recognition, Santa Barbara, California. 1998. P. 232–237.
41. **Chen Y., Davoine F.** Simultaneous Tracking of Rigid Head Motion and Non-rigid Facial Animation by Analysing Local Features Statistically // Proc. British Machine Vision Conf., Edinburgh. 2006. Vol. 2. P. 609.
42. **Basu S., Essa I., Pentland A.** Motion Regularization for Model-Based Head Tracking // In Proc. IEEE Intl. Conf. on Pattern Recognition, Vienna, Austria. 1996. Vol. 3. P. 611–616.
43. **Cascia M., Sclaroff S., Athitsos V.** Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models // IEEE Trans. on Pattern Analysis and Machine Intelligence. Washington, DC, USA: IEEE Computer Society. 2000. Vol. 22. N 4. P. 322–336.

44. **Seo K., Cohen I., You S., Neumann U.** Face Pose Estimation System by Combining ICA-SVM Learning and 3D Modeling // Proc. Asian Conf. on Comp. Vision, Jeju, Korea 2004. P. 105.
45. **Saha S., Shen C., Hsu C., Aggarwal G., Veeraraghavan A., Sussman A., Bhattacharyya S. S.** Model-Based OpenMP Implementation of a 3D Facial Pose Tracking System // Intl. Conf. Parallel Processing Workshops. Los Alamitos, CA, USA: IEEE Computer Society. 2006. P. 66–73.
46. **Parke F.** Computer Generated Animation of Faces // Proc. of ACM Annual Conf., Boston, Massachusetts, United States, 1972. Vol. 1. P. 451–457.
47. **Rydfalk M.** CANDIDE, a Parameterized Face // Report No. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linköping University, Sweden, 1987.
48. **Ahlberg J.** Candide 3 — An Updated Parametrized Face // Report No. LiTHISY-R-2326, Dept. of Electrical Engineering, Linköping University, Sweden. 2001. 16 p.
49. **Anisetti M., Bellandi V., Damiani E., Beverina F.** 3D Expressive Face Model-Based Tracking Algorithm // Proc. 24th IASTED Intl. Conf. on Signal Processing, Pattern Recognition, and Applications, Anaheim, CA, USA: ACTA Press 2006. P. 111–116.
50. **Edwards G. J., Taylor C. J., Cootes T. F.** Interpreting face images using active appearance models // Proc. International Conf. on Automatic Face and Gesture Recognition, 1998. P. 300–305.
51. **Xiao J., Baker S., Matthews I., Kanade T.** Real-Time Combined 2D + 3D Active Appearance Models // Proc. IEEE Conf. on Comp. Vision and Pattern Recognition. 2004. Vol. 2. P. 535–542.
52. **Matthews I., Baker S.** Active Appearance Models Revisited // Intl. Journal of Computer Vision, Hingham, MA, USA: Kluwer Academic Publishers 2004. Vol. 60. N 2. P. 135–164.
53. **Matthews I., Xiao J., Baker S.** 2D vs. 3D Deformable Face Models: Representational Power, Construction and Real-Time Fitting // Intl. Journal of Comp. Vision. Springer. 2007. Vol. 75. N 1. P. 93–113.
54. **Ramnath K.** On the Multi-View Fitting and Construction of Dense Deformable Face Models // Master's thesis, tech. report CMU-RI-TR-07-10, Robotics Institute, Carnegie Mellon University. 2007. 79 p.
55. **Black M., Jepson A.** EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View Based Representation // Intl. Journal Comp. Vision. Springer Netherlands, 1998. Vol. 26. N 1. P. 63–84.
56. **Morency L., Sunberg P., Darrel T.** Pose Estimation Using 3D View-Based Eigenspaces // IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures. Cambridge, MA, USA 2003. P. 45–52.
57. **Wang P., Ji Q.** Multi-View Face Detection Under Complex Scene based on Combined SVMs // 17th Intl. Conf. on Pattern Recogn. Washington, DC, USA: IEEE Computer Society, 2004. Vol. 4. P. 179–182.
58. **Avidan S.** Support Vector Tracking // IEEE Trans. on Patt. Anal. and Mach. Intell. New-York, NY, USA : IEEE Computer Society Vol. 26. N 8. 2004. P. 1064–1072.
59. **Krüeger V., Bruns S., Sommer G.** Efficient Head Pose Estimation with Gabor Wavelet Networks // Proc. BMVC. 2000. P. 12–14.
60. **Krüeger V., Sommer G.** Gabor Wavelet Networks for Efficient Head Pose Estimation // Image and Vision Computing. Amsterdam, Netherlands: Elsevier. 2002. Vol. 20. N 9. P. 665–672.
61. **Wei Y., Fradet L., Tan T.** Head Pose Estimation Using Gabor Eigenspace Modeling // Proc. of Intl. Conf. on Image Processing. 2002. Vol. 1. P. 281–284.
62. **Ma B., Zhang W., Shan S., Chen X., Gao W.** Robust Head Pose Estimation Using LGBP // 18th Intl. Conf. on Pattern Recognition, Hong Kong 2006. Vol. 2. P. 512–515.
63. **Gourier N., Maisonnasse J., Hall D., Crowley J.** Head Pose Estimation on Low Resolution Images // Multimodal Technologies for Perception of Humans. Berlin, Germany: Springer Berlin / Heidelberg 2007. Vol. 4122/2007. P. 270–280.
64. **Zhang Q., Benveniste A.** Wavelet networks // IEEE Trans. Neural Networks 1992. Vol. 3. P. 889–898.
65. **Viola P., Jones M. J.** Robust Real-Time Face Detection // Intl. Journal of Computer Vision. Kluwer Academic Publishers: Netherlands. 2004. Vol. 57. N 2. P. 137–154.
66. **Wu B., Ai H., Huang C., Lao S.** Fast Rotation Invariant Multi-View Face Detection Based on Real Adaboost // In Proc. 6th IEEE International Conference on Automatic Face and Gesture Recognition, 2004. P. 79–84.
67. **An K., Yoo D., Jung S., Chung M.** Robust Multi-view Face Tracking // IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Alberta, Canada. 2005. P. 1905–1910.
68. **Meynet J., Arsan T., Mota J. C., Tiran J.** Fast Multiview Face Tracking with Pose Estimation // Tech. report TR-ITS.2007.01. Ecole Polytechnique Federale de Lausanne, Signal Processing Institute 2007. 12 p.
69. **Huang C., Ai H., Li Y., Lao S.** High-Performance Rotation Invariant Multiview Face Detection // IEEE Trans. on Pattern Analysis and Machine Intelligence, 2007. Vol. 29. N 4. P. 671–686.
70. **Li Y., Ai H., Huang C., Lao S.** Robust Head Tracking with Particles Based on Multiple Cues Fusion // Proc. of CVHCI06. Berlin Heidelberg: Springer-Verlag. 2006. P. 29–39.
71. **Ho J., Lee K., Yang M., Kriegman D.** Visual Tracking Using Learned Linear Subspaces // IEEE Comp. Society Conf on Computer Vision and Pattern Recognition Los Alamitos, CA, USA: IEEE Computer Society, 2004. Vol. 1. P. 782–789.
72. **Chaumont M., Puech W.** 3D-Face Model Tracking Based on a Multi-Resolution Active Search // Visual Communications and Image Processing, Part of IST/SPIE 19th Annual Symposium on Electronic Imaging, San Jose, California, USA 2007. Vol. 6508. P. 65081U.
73. **Xiao J., Georgescu B., Zhou X., Comaniciu D., Kanade T.** Simultaneous Registration and Modeling of Deformable Shapes // In Proc. IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition, New York, NY. 2006. Vol. 2. P. 2429–2436.
74. **Davoine F., Dornaika F.** Head and Facial Animation Tracking using Appearance Adaptive Models and Particle Filters // Comp. Vision and Pattern Recognition Workshop, Los Alamitos, CA, USA: IEEE Computer Society. 2004. Vol. 10. P. 153.
75. **Goldenstein S., Vogler C., Stoff J., Pavlovic V., Metaxas D.** Outlier Rejection in Deformable Model Tracking // Computer Vision and Pattern Recognition Workshop Los Alamitos, CA, USA: IEEE Computer Society 2004. Vol. 1. P. 19.

Е. М. Вахромцев, аспирант,
Московский государственный университет
путей сообщения (МИИТ),
e-mail: e.vakhromtsev@gmail.com

Применение методов "живого" моделирования для производственного мониторинга

Рассматривается уникальный способ первичной диагностики и мониторинга управленческого процесса с помощью методов "живого моделирования", включающих в себя аппарат логических нейронных сетей и современную реализацию скелетной анимации. "Живое" моделирование позволяет представить происходящие в системе процессы в виде трехмерного виртуального персонажа, который наглядно демонстрирует работу системы.

Ключевые слова: "живое" моделирование, реагирующий объект, виртуальный персонаж, скелетная анимация, компьютерная графика, мониторинг, первичная диагностика, логическая нейронная сеть

Введение

Динамику современного производственного (или управленческого) процесса отображает множество постоянно меняющихся показателей. Обычно они выводятся на диспетчерские пульта или мониторы. Оперативное управление требует постоянного контроля всех показателей для формирования целостной картины текущей ситуации. Это помогает принимать управленческие и производственные решения, а также экстренные меры безопасности. Человеческий разум не всегда может отслеживать динамику всех протекающих на производстве процессов, но, тем не менее, необходимо принимать оперативное решение. Возникает идея введения "посредника" — реагирующего объекта, который хотя бы грубо мог отреагировать на появление опасной ситуации, провести грубую диагностику или просто в общих чертах представить текущую ситуацию. Описываемые в статье механизмы могут использоваться в широчайшей области человеческой деятельности, везде, где возникает потребность в предварительном анализе динамически изменяющихся данных. Это — производство, энергетика, транспорт, медицина, бизнес, искусство, сфера развлечений, а также многое другое.

Для решения задачи мониторинга можно воспользоваться принципом "живого" моделирования, который впервые был предложен А. Б. Барским в [1]. "Живое" моделирование использует

нейросетевые технологии для имитации поведения живого существа. Информация о ситуации, благодаря "живому" моделированию, подается в простом и интуитивно понятном виде, доступном даже несведущему человеку. Оценка состояния системы можно получить, бросив беглый взгляд на внешний вид и поведение некоторого виртуального существа.

Не секрет, что многие организации имеют в своем "PR-поле" некоторый талисман, отождествляемый с живым персонажем. Такой персонаж мог бы стать неплохим индикатором отдельных производственных процессов или целостного благополучия компании. Установленная фигура-робот в центральном офисе организации могла бы служить не только информативно-развлекательным, но и рекламным целям. Виртуальное воплощение такого существа могло бы появляться на компьютерах сотрудников или даже сопровождать специализированное корпоративное ПО (вспомните знаменитого помощника-скрепку в популярном офисном пакете).

Постановка задачи "живого" моделирования реагирующего объекта

Идеи "живого" моделирования нашли свое применение в электронной системе планирования и управления проектами, применяемой в издательстве MEDIACRAT. Издательство занимается выпуском собственных и клиентских периодических изданий, различными медийными проектами. Используемое программное обеспечение — это "интеллектуальная" система, автоматизирующая действия рабочей группы, которая осуществляет планирование, разработку и ведение дел издательства. Задача создания реагирующего объекта возникла в связи с ростом числа проектов, что привело к необозримости текущего состояния работ по каждому из них и всех в совокупности. В рабочую группу каждого проекта входят несколько специалистов: бухгалтер, юрист, редактор, художник-дизайнер, специалист по производству, менеджер по подписке и менеджер по связи с заказчиком. Кроме того, задачей управления и организации всех проектов в совокупности занимается координатор проектов. Этот человек владеет максимально полной информацией о каждом проекте, определяет, насколько успешно продвигается проект, где каких ресурсов не хватает или, наоборот, не вполне раскрыт потенциал возможностей производства. Важно, что исходя из текущей ситуации, своего опыта и интуиции координатор выставляет общую оценку успешности проекта. Эта оценка используется ответственно-

ными сотрудниками и руководством компании. Однако при превышении некоторого критического значения числа проектов один координатор перестает справляться со своими обязанностями. Слишком много различных параметров (зачастую трудно формализуемых) и мелких факторов приходится держать в голове. Эту ситуацию можно решить, распределив роли между несколькими координаторами. Но даже в этом случае остается задача координации координаторов. Здесь и получил воплощение способ решения возникшей задачи — это разработка специализированного программного обеспечения, основанного на принципе "живого" моделирования.

Итак, возникла задача мониторинга деятельности предприятия. При этом особенно важна задача оценки происходящих бизнес-процессов. Кроме того, при формулировании технического задания на систему мониторинга руководством компании было высказано желание представить итоговую оценку в максимально доступной и привлекательной форме. Состояние дел должно характеризоваться обликом и поведением виртуального персонажа, который создается с помощью "живого" моделирования.

Развитие средств компьютерной графики для "живого" моделирования

На сегодняшний день компьютерная реализация "живого" моделирования ограничена мультимедийными средствами двумерной и трехмерной графики. Возможности создаваемых этими средствами систем и образов огромны. Однако основной недостаток многих известных систем кроется в априорном знании возможных сцен, сглаживание перехода через которые в реальном времени имитирует движение. Это подобно табличному заданию огромного числа значений функции многих переменных при игнорировании непосредственного расчета.

Непредсказуемость сцен, возникающая при имитации реакции моделируемого объекта на внешние воздействия, может потребовать огромного, практически не реализуемого, числа предполагаемых возможных сцен.

В этом случае желательно при построении объекта (монстра, человека и др.) снабжать его управляемыми органами движения (модификации, деформации, имитирующей улыбку, движение руки и т. д.). Их необходимо соединить с

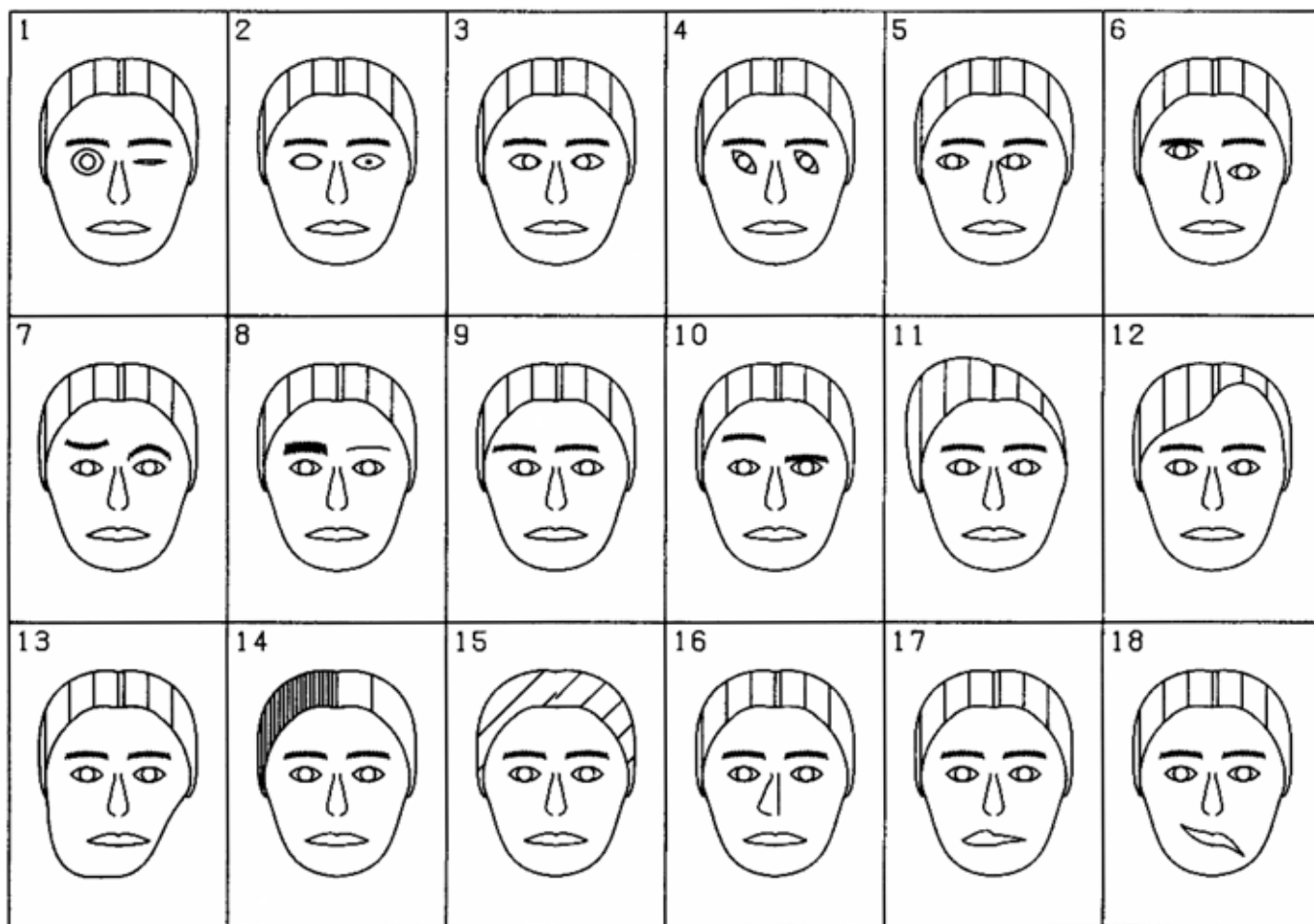


Рис. 1. Иллюстрация лиц Чернова из [7]

управляющим элементом — жестко заданной программой. Такая программа может предполагать анализ ситуации или внешних воздействий. Это напрямую требует применения аппарата логических нейронных сетей [2—5], одновременно и в разной степени реализующих ряд отношений вида "если — то", что наиболее близко к поведению живого существа.

Заметим, что проблемы компьютерной реализации "живого" моделирования целиком включают проблемы натурального моделирования. При желании компьютерная графика может быть заменена робототехникой.

Различные способы наглядного представления мультимедийных данных уже достаточно долго привлекают к себе внимание исследователей. Одним из таких способов является метод "лиц Чернова" (Н. Chernoff) [6, 7]. Облик лиц Чернова может быть описан восемнадцатью параметрами, такими как размер, положение и наклон глаз, носа, ушей и других черт лица. Простоту и наглядность этого метода легко понять, взглянув на рис. 1.

Использование и развитие скелетной анимации трехмерного персонажа

В [1] для визуализации персонажа А. Б. Барский предлагает применять технику трехмерной памяти и механизм мышц для визуализации оживляемого объекта. Эти методы достаточно универсальны, но их реализация на ПК не может быть оптимальной, так как не позволяет задействовать современные аппаратные средства для работы с трехмерной графикой. Современные графические ускорители рассчитаны на работу с полигональной графикой. Развитие графических процессоров идет в сторону повышения универсальности их использования, но в настоящее время наиболее зрелищных и производительных сцен можно достигнуть только с помощью полигональной трехмерной графики. Очевидно, что для реализации "живого" моделирования необходимы возможности трехмерной графики реального времени. Современные средства написания программ включают в себя стандартные 3D API, из которых самыми востребованными являются OpenGL или DirectX. Эти технологии хорошо зарекомендовали себя в сфере графического 3D-моделирования, виртуальной реальности и компьютерных игр.

Важным требованием к приложению является кроссплатформенность — работа во всех распространенных операционных системах на настольных компьютерах. Поэтому для программы была выбрана технология OpenGL, доступная на всех современных операционных системах и аппаратных платформах.

OpenGL предоставляет только низкоуровневые функции для работы с трехмерной графикой, такие как построение треугольника или линии. Так что весь необходимый графический функционал необходимо разрабатывать заново.

Для динамического изменения облика и поведения персонажа оказалось необходимым предусмотреть два основных типа действий:

- проигрывание заранее подготовленной анимации, например, когда персонаж начинает задумчиво ходить по кругу или танцевать;
- изменение характеристик внешнего облика трехмерной модели, таких как размер и форма глаз или цвет волос.

Комбинация этих двух типов может достаточно хорошо отражать состояние контролируемых показателей. Возможный облик персонажа, выбранный для программы, приведен на рис. 2 (см. вторую сторону обложки).

Для обеспечения анимации трехмерного персонажа могут быть использованы различные методы. Самый простой и классический — покадровая анимация [8], при которой для каждого кадра задается фиксированное расположение вершин. Этот метод прост в реализации, но обладает рядом недостатков. Во-первых, покадровая анимация не может быть использована для реализации динамической анимации, когда поведение трехмерной модели динамически управляется некоторой функцией. Во-вторых, покадровая анимация требовательна к оперативной памяти.

Частично решить проблему потребления оперативной памяти и недостаточной "гибкости" может так называемая анимация по ключевым кадрам (или анимация с интерполяцией, иногда технологию называют твининг (*tweening*)) [9]. Этот метод заключается в том, что для каждой вершины модели хранятся ее координаты для некоторого набора моментов времени (ключевых кадров) t_0, \dots, t_n :

$$v_{i,j} = v_i(t_j),$$

где $j = 1, \dots, n$.

В этом случае для вычисления координаты вершины в некоторый момент времени t выбираются два соседних ключевых кадра, $t_i \leq t \leq t_{i+1}$. Затем с помощью интерполяции (как правило, линейной или сплайновой) между координатами точки в этих кадрах находятся требуемые координаты вершины в момент времени t .

Такой способ достаточно прост и не так требователен к аппаратным ресурсам, но все же не решает проблему повышенного потребления памяти до конца, а также не обеспечивает достаточной гибкости в управлении моделью. Поведение модели остается ограниченным ключевыми кадрами.

Для преодоления этих трудностей можно воспользоваться простым и элегантным способом — скелетной анимацией, иногда называемым методом шарнирных фигур [8, 9].

Метод скелетной анимации не обладает перечисленными выше недостатками. Фактически для анимации хранится только "скелет", т. е. набор векторов (или трехмерных преобразований), к которым привязываются все вершины модели. Каждая вершина может быть привязана к одной или нескольким "костям" (т. е. векторам "скелета"). Размер самого скелета (число "костей") очень мал по сравнению с числом вершин модели, "натягиваемой" на скелет. Пример скелета показан на рис. 3 (см. третью сторону обложки). Скелетная анимация позволяет динамически управлять моделью, что необходимо для "живого" моделирования.

Теперь достаточно анимировать только скелет, и положение скелета автоматически обеспечит корректные координаты всех вершин модели. Именно скелет отвечает за динамику модели, и одному скелету может соответствовать несколько различных моделей (например, один и тот же скелет может "анимировать" несколько разных моделей людей). Важно, что использование скелета позволяет не только экономить память для анимации, но и динамически анимировать персонаж путем изменения положения в пространстве его костей.

Покадровая анимация с интерполяцией или без нее по-прежнему может применяться, но уже только для скелета, а не для всей модели. Это позволяет существенно сократить объем используемой оперативной памяти. Трехмерная модель виртуального персонажа может состоять из тысяч и даже сотен тысяч вершин. Для скелета, как правило, достаточно десятков-сотен вершин.

Скелет в простейшем случае представляет собой набор "костей", соединенных иерархическим образом. На рис. 4 приведен схематичный скелет руки.

Каждая кость скелета — это отрезок, начало которого подсоединено к концу "родительской кости". В более сложных случаях к соединениям костей могут быть добавлены ограничивающие условия. Это может пригодиться для того, чтобы руки модели-человека не выворачивались в неестественное положение. Так же можно добавить мышцы — траектории, суперпозиция которых задает положение костей скелета. Мышцы хороши тем, что позволяют управлять трехмерной моделью человека путем задания уровня напряженности определенных групп мышц. Тем не менее, эти требования не являются обязательными для успешной реализации скелетной анимации.

Для однозначного задания кости используется указатель на родительскую кость, а также конец

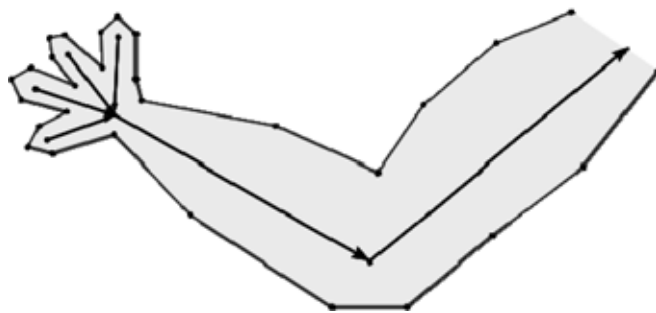


Рис. 4. Упрощенное представление руки со скелетной основой внутри

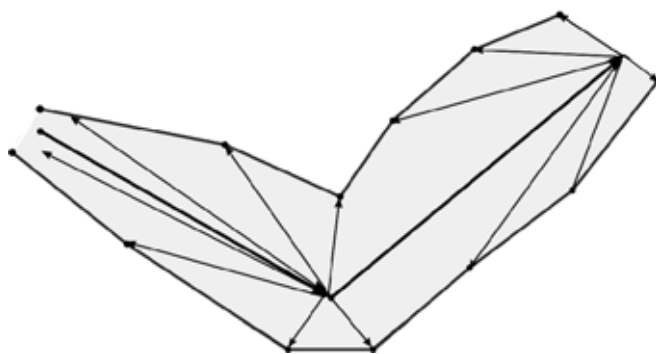


Рис. 5. Задание вершин относительно костей скелета

отрезка текущей кости. Конец отрезка обычно задается как аффинное преобразование, не изменяющее длины, переводящее начало кости в ее конец. Можно показать, что такое преобразование всегда представляет собой суперпозицию поворота и переноса.

Поворот и перенос для каждой кости задаются относительно конца ее родительской кости. Однако в целом ряде случаев (например, при построении вершин анимируемой модели) необходимо знать преобразование, задаваемое костью относительно системы координат модели, т. е. корня скелета. Для этого параметры родительской кости должны быть доступны для всего скелета. Если для хранения костей используется массив, в котором для любой кости родительская кость располагается в массиве раньше самой кости, то первая кость в массиве является корневой.

Для создания связи между скелетом и вершинами модели в простейшем случае каждая вершина сопоставляется определенной кости, относительно конца которой и задается вершина (рис. 5).

Чтобы найти координаты вершины по скелету модели, достаточно найти абсолютное преобразование, определяемое требуемой костью и вычислить с его помощью координаты.

Когда необходимо найти не только положение вершины, но и связанные с моделью направления (нормаль, касательный вектор и т. д.), то строится



Рис. 6. Персонаж с обозначенным скелетом

абсолютное преобразование для кости, состоящее из поворота и переноса.

Сама вершина преобразуется с использованием как поворота, так и переноса, а векторы направлений преобразуются только с помощью поворота.

Иногда используется и более гибкая схема — для каждой вершины задается не одна кость, а несколько. Для каждой из таких костей кроме положения вершины относительно нее задается также и значимость данной кости. Результирующее положение вершины берется как взвешенная сумма преобразованных костями вершин. Для любой вершины сумма всех весов должна равняться единице.

Это позволяет при анимации вершины учитывать сразу несколько костей, что дает возможность создания более реалистичной анимации.

Таким образом, все анимации персонажа (в том числе и получаемые путем динамических расчетов для скелета) сводятся к анимации самого скелета. На рис. 6 показан скелет для используемой трехмерной модели.

Динамическое управление реагирующим объектом

Поведение персонажа основывается на вводимых пользователями данных. В процессе работы оператор (координатор проектов) вводит в программу данные, несущие как информативную, так и оценочную информацию. Например, для орга-

низации-подрядчика координатор вводит адрес и контактные данные компании, а также общую оценку-статус этой компании, такую как "ответственный партнер" или "безответственный исполнитель". Фактически программа предлагает пользователю оценить ответственность партнера по некоторой цифровой шкале. Исходя из предложенных вариантов программа выбирает некоторый положительный или отрицательный коэффициент-оценку данной организации. Важно, что кроме координатора в программе работают и другие сотрудники, которые тоже могут выставлять свои оценки. Оценки сотрудников не равнозначны. Вес оценки зависит от должности, опыта и других характеристик сотрудника, определяемых руководством и/или координатором проектов. Кроме того, оценки могут иметь некоторую "инерцию" за счет обобщения выставляемых ранее оценок. С течением времени оценки "устаревают", коэффициент их влияния на конечную оценку постепенно уменьшается до нуля.

Программа также может выставлять свою оценку. Система проверяет выполнение поставленных условий (выполнены ли работы в срок, не изменилась ли неожиданно цена поставляемых материалов, число успешно выполненных заданий) и исходя из этого выставляет свой коэффициент-оценку.

Основными сущностями в программе являются организации, сотрудники (как собственные, так и сторонних организаций) и проекты, делящиеся на этапы выполнения.

Для оживления персонажа используется логическая нейронная сеть. Настройка параметров-весов нейросети подбирается эмпирически во время тестирования и настройки программы, в процессе эксплуатации. Разработанная программа настраивается путем задания связей между

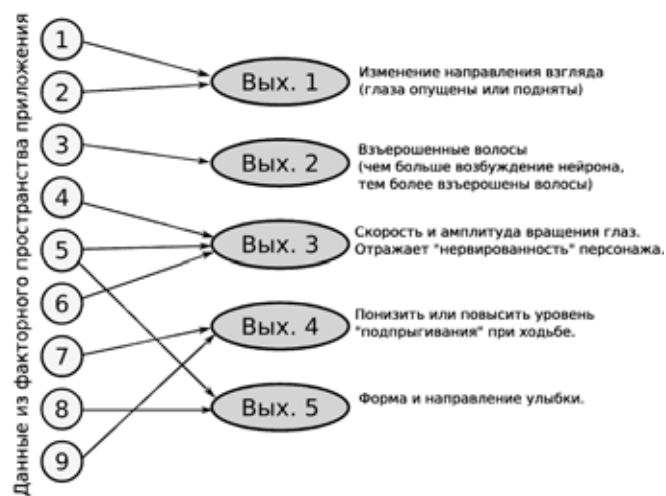


Рис. 7. Иллюстрация работы нейросети, управляющей поведением персонажа

входными показателями ("датчиками") и анимационными функциями виртуального персонажа. Датчики — это формователи сигналов из программных данных. Настроенные связи между датчиками и анимационными функциями персонажа преобразуются программой в однослойную логическую нейронную сеть, на рецепторы которой подается информация, соответствующая достоверности принадлежности данных к некоторому значению или интервалу значений.

Выходной слой нейронной сети управляет состоянием и поведением объекта. Например, размер существа-эмблемы может зависеть от предварительной оценки (ожиданий) управляющих партнеров, а с приближением сроков сдачи персонаж начинает сильнее нервничать. На рис. 7 изображена схематическая логическая нейронная сеть для управления поведением персонажа.

Выводы. Реализация и развитие

Достижения в области компьютерной графики значительны. Однако постановка задачи "живого" моделирования требует решения проблемы динамической связи управляющего элемента с графическим объектом. Разработка средств такого взаимодействия с помощью логической нейронной сети отражена в настоящей статье.

В жизни нас окружает множество трудно формализуемых задач. Одна из таких задач — первичная оценка происходящего. Предложенный метод первичной диагностики на основе "живого" моделирования не отвергает привычных методов мониторинга или оценки систем и процессов, а дополняет их. Обнаружив изменение (а возможно, и осложнение) ситуации, можно перейти к детальному изучению специфичных для данной системы параметров обычными средствами (просматривать показания сложных приборов, проводить диагностики, экспертизы, совещания и т. п.). В качестве эксперимента были разработаны несколько прикладных программных систем-посредников, которые проводят общую диагностику и по возможности выполняют какие-то действия или просто информируют пользователя о смене ситуации. Программы используют принципы со-

временной трехмерной графики, которые широко распространены в компьютерных играх, пакетах трехмерного моделирования и системах виртуальной реальности. Современные аппаратные средства позволяют осуществлять качественную и зрелищную анимацию. Благодаря тому, что информация представляется в неформальном, игровом виде, многие пользователи более благосклонно воспринимают предложенные программы и охотнее ими пользуются. Кроме того, не стоит пренебрегать так называемым "вау-эффектом", когда пользователь бывает поражен способом взаимодействия с программой как с живым объектом, а не с бездушной системой.

Известны случаи, когда врач по одному только внешнему виду пациента может определить болезнь. Использование "живого" моделирования позволяет представить сложную систему со множеством исходных данных в простом и доступном виде. При этом, чтобы понять, что "что-то случилось", не нужно быть специалистом и глубоко разбираться в предложенной системе, достаточно бросить беглый взгляд на виртуального персонажа, представляющего эту систему. А при определенном навыке и опыте наблюдений за объектом можно сразу сказать, в какой области обнаружены изменения или неполадки.

Список литературы

1. **Барский А. Б.** Логические нейронные сети и "живое" моделирование // Компьютеры в учебном процессе. 2005. № 11.
2. **Барский А. Б.** Нейронные сети и искусственный интеллект // Информационные технологии. Приложение. 2003. № 1. 32 с.
3. **Барский А. Б.** Нейронные сети: распознавание, обучение, принятие решений. М.: Финансы и статистика, 2004.
4. **Барский А. Б.** Нейроинформатика. Однослойные логические нейронные сети. М.: МИИТ, 2008.
5. **Барский А. Б.** Логические нейронные сети для систем управления и принятия решений // Нейрокомпьютеры: разработка и применение. 2005. № 6.
6. **Chernoff H.** The Use of Faces to Represent Points in K-Dimensional Space Graphically // Journal of the American Statistical Association. 1977. Vol. 68. N 342. P. 361—368.
7. **Flury B., Riedwyl H.** Graphical Representation of Multivariate Data by Means of Asymmetrical Faces // Journal of the American Statistical Association. 1981. Vol. 76. N 376.
8. **Боресков А. В.** Основы скелетной анимации. (<http://steps3d.narod.ru/tutorials/skeletal-animation-tutorial.html>)
9. **Херн Д., Паулин Бейкер М.** Компьютерная графика и стандарт OpenGL. М.: Издат. дом "Вильямс", 2005.

УДК 519.622.2

Д. М. Жук, канд. техн. наук, доц.,
В. Б. Маничев, канд. техн. наук, доц.,
А. О. Ильницкий, аспирант,
МГТУ им. Н. Э. Баумана,
e-mail: manichev@bmstu.ru

Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временной области. Часть 1

Приведены результаты анализа и обоснования выбора базовых методов интегрирования нелинейных систем дифференциально-алгебраических уравнений (ДАУ) при разработке нового универсального, высоконадежного программно-технического комплекса FMS-PA10 (Framework for Mathematical Simulation, Program for Analysis version 10) для моделирования систем и объектов во временной области. На примере неявного метода трапеций разработан новый комбинированный корректирующий алгоритм для реализации АЛ-устойчивых методов интегрирования систем ДАУ, которые будут положены в основу при разработке программно-технического комплекса FMS-PA10.

Ключевые слова: моделирование, обыкновенные дифференциальные уравнения, дифференциально-алгебраические уравнения, методы интегрирования, временная область, линейные алгебраические уравнения

Введение

В статье [2] были выполнены анализ и обоснование выбора обобщенного базиса переменных для моделирования систем и объектов во временной области в качестве основы для разработки программно-технического комплекса для моделирования систем и объектов, описываемых системами дифференциально-алгебраических уравнений (ДАУ), — нового универсального, высоконадежного программно-технического комплекса FMS-PA10 (Framework for Mathematical Simulation, Program for Analysis version 10). Предполагалось, что в FMS-PA10 будут использованы новые неявные АЛ-устойчивые методы интегрирования систем ДАУ (абсолютно (А-) устойчивые строго в левой (Left) полуплоскости области абсолютной

устойчивости методов интегрирования) с порядком точности выше второго, а именно неявный АЛ-устойчивый метод четвертого порядка точности [3—5]. Однако решение тестовых задач, имеющих многопериодное решение, показало, что для данного класса задач необходимо использовать АЛ-устойчивые методы интегрирования более высокого порядка точности (6, 8, 10-го и т. д.). Кроме того, в АЛ-устойчивых методах высокого порядка точности, так же как и в неявном методе трапеций, возникла проблема "ложных колебаний".

В данной работе дается обоснование выбора в качестве базовых методов интегрирования систем ДАУ для программно-технического комплекса FMS-PA10 неявного метода Эйлера и библиотеки АЛ-устойчивых методов интегрирования 2, 4, 6-го и т. д. порядка точности. Кроме того, предлагается и обосновывается новый алгоритм для решения проблемы "ложных колебаний" в неявном методе трапеций — комбинированный корректирующий алгоритм реализации неявного метода трапеций, основанный на идее комбинирования явной и неявной формул Эйлера [1]. Этот алгоритм будет применен и для АЛ-устойчивых методов интегрирования более высокого порядка точности.

Выбор методов интегрирования систем ДАУ при моделировании систем и объектов во временной области

Моделирование систем и объектов во временной области сводится в наиболее общем случае к решению систем ДАУ, неразрешенных относительно производных и представленных в виде

$$F(X, XP, Y, t) = 0, \quad (1)$$

где X — вектор дифференцируемых переменных размерностью m ; XP — вектор производных этих переменных по переменной t размерностью m , т. е. $XP = dX/dt$; Y — вектор алгебраических переменных размерностью k ; t — независимая переменная (обычно — время); F — вектор-функция размерностью l , где $l = m + k$. Заданы начальные условия $X_0 = X(0)$ и отрезок интегрирования.

В теоретическом плане свойства систем обыкновенных дифференциальных уравнений (ОДУ) и методов их решения рассматриваются применительно к линейным неоднородным системам ОДУ вида

$$dX/dt = AX + G(t), \quad (2)$$

где A — постоянная действительная матрица размерностью $m \times m$; $G(t)$ — известная вектор-функция времени.

Если метод интегрирования будет непригоден для решения системы (2), то он будет непригоден и для решения системы (1). Обозначим определитель матрицы A через $\det(A)$. Спектр собственных значений — это множество всех собственных значений λ_i матрицы $A, i = 1, 2, \dots, m$. В общем случае λ_i — комплексные величины, т. е. $\lambda_i = \text{Re}(\lambda_i) + j\text{Im}(\lambda_i)$, где $j = \sqrt{-1}$. Если собственные значения λ_i различны, то аналитическое решение системы (2) можно представить в виде суммы отдельных фундаментальных решений однородной системы ОДУ с матрицей A и частного решения неоднородной системы (2):

$$X(t) = \sum_{i=1}^m c_i e^{\lambda_i t} X_i + Z(t), \quad (3)$$

где c_i — постоянные величины; X_i — собственный вектор матрицы A , соответствующий собственному значению λ_i ; $Z(t)$ — частное решение системы (2).

Характер фундаментальных решений системы ОДУ (2) полностью определяется собственными значениями матрицы A . Система ОДУ будет неустойчивой, если хотя бы для одного i имеем $\text{Re}(\lambda_i) > 0$. При этом i -я компонента в (3) будет неограниченно возрастать с увеличением t . Система ОДУ будет устойчивой, если $\text{Re}(\lambda_i) < 0$ для всех i , так как все фундаментальные решения будут иметь затухающий характер с увеличением t . Система ОДУ будет иметь многопериодное решение, если $\text{Im}(\lambda_i) \gg \text{Re}(\lambda_i)$ для некоторых i .

В зависимости от характера фундаментальных решений можно выделить пять основных типов задач анализа переходных процессов, показанных на рис. 1—6 и связанных с решением систем ОДУ вида (2).

Устойчивость методов интегрирования связана с характером изменения накопленной погрешности:

- если накопленная погрешность в ходе интегрирования не возрастает с увеличением числа шагов интегрирования, то используемый метод интегрирования будет численно устойчивым;
- если при небольших погрешностях аппроксимации и вычислений погрешность накопления растет от шага к шагу, то используемый метод интегрирования будет численно неустойчивым при данных шагах интегрирования.

Исследование устойчивости методов интегрирования систем ОДУ и ДАУ выполняется путем решения одного модельного уравнения с постоянным шагом интегрирования h при $t \rightarrow \infty$ и сравнения численного решения с аналитическим [7—8]: $dx/dt = \lambda x, x_0 = 1, \lambda$ — комплексный коэффициент.

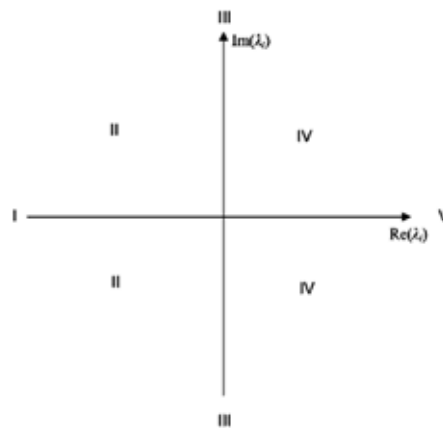


Рис. 1. Области устойчивости для пяти основных типов задач (I—V) при решении систем ОДУ

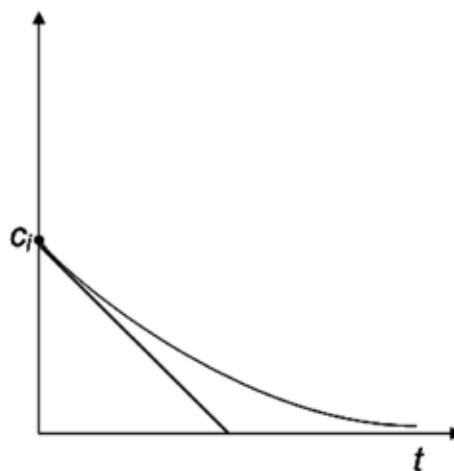


Рис. 2. Переходный процесс i -го фундаментального решения для задач типа I — фундаментальные решения лежат в основном в области I

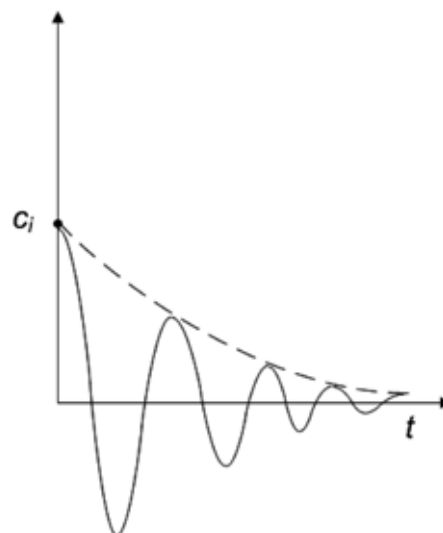


Рис. 3. Переходный процесс i -го фундаментального решения для задач типа II — фундаментальные решения лежат в основном в области II

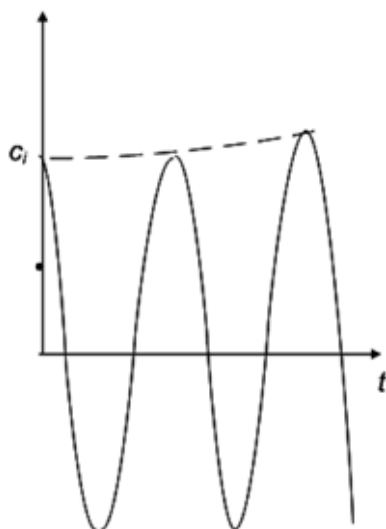


Рис. 4. Переходной процесс i -го фундаментального решения для задач типа III — фундаментальные решения лежат в основном в области III

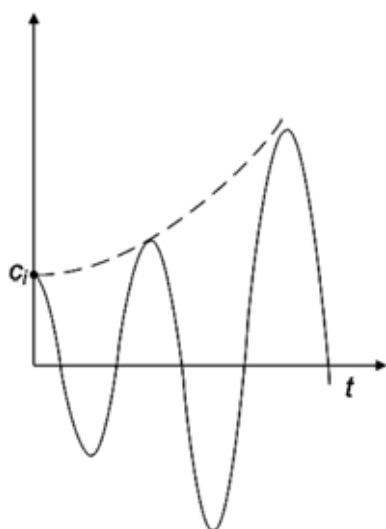


Рис. 5. Переходной процесс i -го фундаментального решения для задач типа IV — фундаментальные решения лежат в основном в области IV

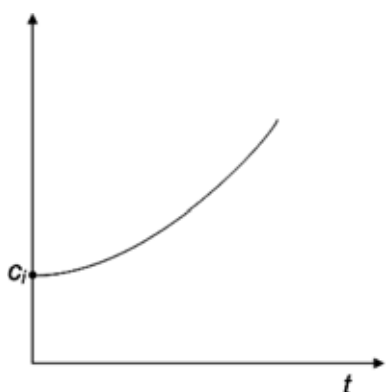


Рис. 6 Переходной процесс i -го фундаментального решения для задач типа V — фундаментальные решения лежат в основном в области V

Метод интегрирования называется абсолютно устойчивым, или А-устойчивым, для заданного значения $h\lambda$, если численное решение модельного уравнения при $t \rightarrow \infty$ является невозрастающей величиной. Абсолютная устойчивость является естественным требованием для задач типа I и II, рассмотренных выше, т. е. при $\text{Re}(h\lambda) < 0$, так как в этом случае модуль аналитического решения модельного уравнения — также невозрастающая функция.

Поскольку явные методы интегрирования систем ОДУ и ДАУ не являются универсальными и не отвечают требованиям устойчивости при решении с переменным шагом интегрирования жестких систем дифференциальных уравнений [4–8], то в качестве базовых методов интегрирования для моделирования разнообразных систем и объектов во временной области следует выбирать неявные методы [4–8]. Неявные методы интегрирования ОДУ являются многошаговыми и сводятся к итерационному решению систем нелинейных алгебраических уравнений. При программной реализации этих методов следует выделить два основных типа погрешностей численного решения систем ОДУ: глобальная погрешность (или качественная ошибка) и локальная погрешность (или количественная ошибка). Программисты основное внимание уделяют получению гарантированной математической локальной точности, хотя главное, конечно, — это получение качественно корректного численного решения. Чтобы гарантировать получение качественно корректного решения метод интегрирования должен быть устойчивым для устойчивых систем ОДУ и должен быть неустойчивым для неустойчивых систем ОДУ, поэтому методы интегрирования должны быть АL-устойчивыми, т. е. абсолютно устойчивыми строго в левой полуплоскости комплексной плоскости устойчивости, как показано на рис. 7.

В общем случае при решении разнообразных систем ДАУ вида (1) можно выделить шесть основных типов задач, требующих применения различных неявных методов интегрирования.

1. Решение любых систем нелинейных алгебраических уравнений методом дифференцирования по параметру или расчет статических режимов моделируемых устойчивых динамических объектов при не зависящих от времени параметрах системы ДАУ путем прямого интегрирования системы ДАУ от произвольных (чаще нулевых) начальных условий. Отрезок интегрирования выбирается от 0 до 1 (в методе дифференцирования по параметру) или из условия завершения всех переходных процессов в интегрируемой системе ДАУ. Основное требование к методу интегрирования — А-устойчивость, требования по точности не предъявляются (траектории выхода в устойчи-

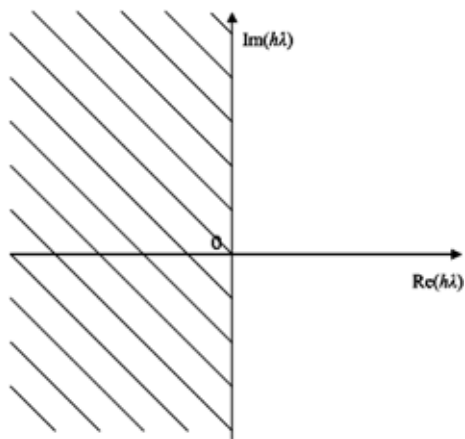


Рис. 7. Область A-устойчивости для AL-устойчивых методов интегрирования

вое состояние могут быть любыми). Для задач этого типа лучшим будет применение только неявного метода Эйлера первого порядка точности без контроля локальной погрешности интегрирования.

2. Расчет устойчивых систем ДАУ с преобладанием аperiodических динамических процессов. Основное требование к методу интегрирования — A-устойчивость. Если к точности интегрирования предъявляются невысокие требования (когда погрешности в определении параметров системы ДАУ достаточно большие и требуется оценить качественно характер динамических процессов), то лучшим будет неявный метод Эйлера. При высоких требованиях к точности интегрирования лучшими для данного класса задач считаются неявные методы формул дифференцирования назад (ФДН) с переменным шагом и порядком точности интегрирования (эти методы в настоящее время используются наиболее широко) по умолчанию в таком известном комплексе для инженерного анализа механических систем, как ADAMS: GSTIFF (метод интегрирования Гира—Гупта (Gear—Gupta stiff method)), а также неявный метод трапеций. Такие задачи возникают, например, в том случае, когда требуется с высокой точностью определить момент времени достижения какой-либо переменной заданного порогового значения, при расчете задач небесной механики, при расчете чувствительности выходных параметров моделируемого объекта, полученных на основе решения систем ДАУ, по отношению к изменению внутренних параметров моделируемого объекта и т. п.

3. Расчет устойчивых систем ДАУ с преобладанием периодических динамических процессов. Основное требование к методу интегрирования — A-устойчивость. Лучшими будут неявные методы ФДН, а также неявный метод трапеций и AL-устойчивые неявные методы 4-, 6-, 8-, 10-го и т. д. порядков точности.

4. Расчет систем ДАУ с многопериодным, слаботухающим или незатухающим решением, а также расчет систем ДАУ, которые являются периодически неустойчивыми (например, при моделировании автоколебательных систем). Основное требование к методу интегрирования — AL-устойчивость и высокий порядок точности [3]. При умеренных требованиях к точности интегрирования единственным выбором будет неявный метод трапеций. При высоких требованиях к точности интегрирования, а также в тех случаях, когда требуется с высокой точностью отобразить период и фазу колебаний, необходимо применять AL-устойчивые неявные методы 4-, 6-, 8-, 10-го и т. д. порядков точности или разрабатывать специальные методы интегрирования для получения огибающих траекторий для многопериодного решения [3].

5. Расчет неустойчивых систем ДАУ с преобладанием периодических динамических процессов. Основное требование к методу интегрирования — AL-устойчивость, поэтому лучшими будут неявный метод трапеций и AL-устойчивые неявные методы 4-, 6-, 8-, 10-го и т. д. порядков точности.

6. Расчет неустойчивых систем ДАУ с преобладанием аperiodических динамических процессов (например, при моделировании взрыва). Основное требование к методу интегрирования — AL-устойчивость. Лучшим будет неявный метод трапеций и AL-устойчивые неявные методы 4-, 6-, 8-, 10-го и т. д. порядков точности.

Неявные методы ФДН (эти методы используются по умолчанию в таком известном программном комплексе для инженерного анализа механических систем, как ADAMS [12]) являются одними из наиболее эффективных методов для решения систем ДАУ с преобладанием аperiodических процессов в решении, однако эта эффективность в сильной степени зависит от алгоритмической реализации этих методов, так как они имеют тенденцию к неустойчивости при сильном изменении шагов интегрирования [4, 6—8]. Анализ абсолютной устойчивости многошаговых методов интегрирования дифференциальных уравнений, к которым относится метод ФДН, показывает, что свойству A-устойчивости удовлетворяют методы, порядок точности которых не выше второго, причем среди таких методов наименьшую погрешность аппроксимации имеет только неявный метод трапеций [6].

Таким образом, в качестве базовых методов интегрирования для моделирования динамики разнообразных систем и объектов следует выбрать *неявный метод Эйлера, неявный метод трапеций и AL-устойчивые неявные методы 4-, 6-, 8-, 10-го и т. д. порядков точности.*

Задачи, которые требуют применения методов интегрирования повышенного порядка точности, возникают, например, в том случае, когда требуется с высокой точностью определить момент времени достижения какой-либо переменной заданного порогового значения, при расчете задач небесной механики, при расчете чувствительности выходных параметров моделируемого объекта, полученных на основе решения систем ДАУ, по отношению к изменению внутренних параметров моделируемого объекта и т. п.

Условие абсолютной устойчивости неявного метода Эйлера для модельного уравнения приводит к оценке

$$|(1 + h\lambda) < 1|.$$

Неявный метод Эйлера дает устойчивое решение в левой полуплоскости комплексной плоскости устойчивости $h\lambda$ для всех $\text{Re}(h\lambda) < 0$ и в большей части правой полуплоскости комплексной плоскости $h\lambda$, т. е. этот метод является А-устойчивым (но не является АL-устойчивым), и он идеально подходит для расчета статических состояний моделируемых систем и объектов, а при невысоких требованиях к точности подходит и для расчета динамических процессов в задачах с апериодическим характером решений. В связи с этим неявный метод Эйлера может быть использован при получении решения систем ДАУ для задач типа 1 и 2, рассмотренных выше.

Условие абсолютной устойчивости неявного метода трапеций для модельного уравнения приводит к оценке

$$|(1 + h\lambda/2)/(1 - h\lambda/2)| < 1.$$

Это условие выполняется строго в левой полуплоскости комплексной плоскости $h\lambda$ для всех $\text{Re}(h\lambda) < 0$ и не выполняется строго в правой полуплоскости для всех $\text{Re}(h\lambda) > 0$, т. е. метод трапеций является АL-устойчивым. В связи с этим неявный метод трапеций может быть использован при получении решения систем ДАУ для задач типа 2—6, рассмотренных выше.

Вместе с тем, при $\text{Re}(\lambda) < 0$ и $|\text{Re}(h\lambda/2)| \gg 0$ вместо быстро затухающего аналитического решения модельного уравнения $x_n = e^{nh\lambda}$ (n — номер шага интегрирования) получим следующее численное решение этого уравнения неявным методом трапеций:

$$x_n = (1 + h\lambda/2)^n / (1 - h\lambda/2)^n.$$

Это решение колеблется от +1 до -1 при изменении n . Такое явление получило название "ложных колебаний" ("ringing" по Kundert K. [9]) для неявного метода трапеций.

"Ложные колебания" приводят к неверным решениям для алгебраических переменных системы

ДАУ и имеют место для всех АL-устойчивых методов интегрирования.

Однако неявный метод трапеций обладает следующим важным свойством: при $\lambda = j\text{Im}(\lambda)$ (модель идеального осциллятора, которой соответствует незатухающее осциллирующее решение модельного уравнения с этим значением параметра λ) устойчивому аналитическому решению модельного уравнения $|y(t_n)| = |e^{j\text{Im}(\lambda)nh}| = 1$ соответствует устойчивое численное решение модельного уравнения неявным методом трапеций

$$|y_n| = |1 + j(h\text{Im}(\lambda)/2)|^n / |1 - j(h\text{Im}(\lambda)/2)|^n = 1.$$

Это свойство имеет место для всех АL-устойчивых методов интегрирования. В связи с этим неявный метод трапеций и другие АL-устойчивые методы интегрирования идеально подходят для получения решения задач типа 3—5, особенно для задач со слаботухающими сильно осциллирующими компонентами в решении.

Комбинированный корректирующий алгоритм для реализации неявного метода трапеций и особенности программной реализации неявных методов

Повысить эффективность неявного метода трапеций удалось путем реализации неявного метода трапеций с помощью комбинированного циклического алгоритма [11], основанного на явной и неявной формулах Эйлера. Обозначим h_{ne} и h_{ni} соответственно значения явного и неявного шагов численного интегрирования по методу Эйлера на n -м шаге цикла алгоритма, а $h_n = h_{ne} + h_{ni}$ — значение n -го шага цикла интегрирования, т. е. $h_n = t_n - t_{n-1}$.

В данном алгоритме для начала интегрирования требуется выполнить для $n = 0$ нулевой шаг h_0 при заданных начальных условиях $X_0 = X(0)$ по неявной формуле Эйлера. Шаг $h_0 = h_{0i}$ должен быть на много порядков меньше, чем основные шаги h_n , чтобы не потерять требуемую точность интегрирования.

Комбинированный циклический алгоритм для выполнения n -го шага h_n цикла интегрирования состоит из двух основных этапов ($n = 1, 2, \dots$).

Этап 1. Выполнение явного шага $h_{ne} = h_{(n-1)i}$. Полученные значения векторов X_{ne}, XP_{ne}, Y_{ne} являются исходными для выполнения неявного шага.

Этап 2. Выполнение неявного шага, выбираемого в соответствии со стратегией автоматического выбора шага интегрирования. Полученное значение вектора X_{ni} является решением $X_n = X_{ni}$ системы (1) на n -м шаге цикла интегрирования, при этом $XP_n = XP_{ni}$ и $Y_n = Y_{ni}$.

Неявный шаг выполняется путем решения относительно векторов X_{ni} , XP_{ni} , Y_{ni} системы нелинейных алгебраических уравнений вида

$$\begin{cases} XP_{ni} = (X_{ni} - X_{ne})/h_{ni}; \\ F(X_{ni}, XP_{ni}, Y_{ni}, t_n) = 0. \end{cases}$$

Обязательным условием работы алгоритма является равенство $h_{ne} = h_{(n-1)i}$, поэтому для выполнения явного шага очередного цикла алгоритма все необходимые данные получаются после неявного шага, тогда явный шаг выполняется по формулам

$$\begin{aligned} X_{ne} &= 2X_{(n-1)i} - X_{(n-1)e}; \\ XP_{ne} &= 2XP_{(n-1)i} - XP_{(n-1)e}; \\ Y_{ne} &= 2Y_{(n-1)i} - Y_{(n-1)e}. \end{aligned}$$

Тестирование данного алгоритма показало, что все основные свойства неявного метода трапеций сохраняются при значительном ослаблении "ложных колебаний". В результате для оценки погрешности интегрирования применим способ полиномиальной аппроксимации [1] (в классическом методе трапеций для оценки погрешности интегрирования на каждом шаге интегрирования применяется более трудоемкий метод двойного расчета с половинным шагом интегрирования — метод Рунге [6–8]). Рассмотренный алгоритм соответствует неявному методу трапеций по точности и устойчивости и может быть использован в качестве базового алгоритма для реализации неявного метода трапеций, однако полностью "ложные колебания" этот алгоритм не устраняет. Дальнейшее развитие данного алгоритма (*комбинированный корректирующий алгоритм*) позволило решить проблему "ложных колебаний".

Решение большого числа практических задач показало, что "ложные колебания" появляются только при интегрировании функций, имеющих разрывы производных этих функций по времени (в основном при наличии кусочно-линейных функций, зависящих от времени). Для разработки нового алгоритма, устраняющего "ложные колебания", была выбрана тестовая задача Кокина С. А.: численное интегрирование дифференциальных уравнений емкостного делителя, когда одна из емкостей является функцией напряжения на обкладках соответствующего конденсатора (рис. 8).

На рис. 8 $C1$ — постоянная емкость, равная 1 (не зависит от напряжения на обкладках), $C2$ — выражена функцией $C2 = C1(0,5 - U_{C2})$, удобной для получения аналитического решения.

Для удобства выбран пилообразный входной сигнал от 0 до 1 В (толстые линии на рис. 9), подаваемый от источника напряжения V с периодом $t_1 = 2$ с.

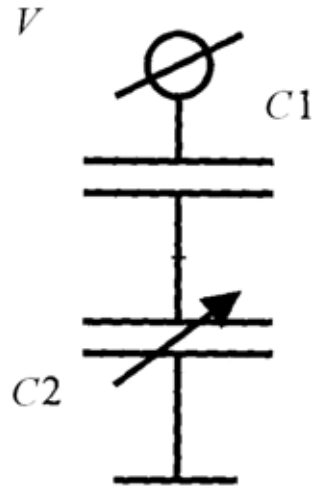


Рис. 8. Тестовая задача Кокина

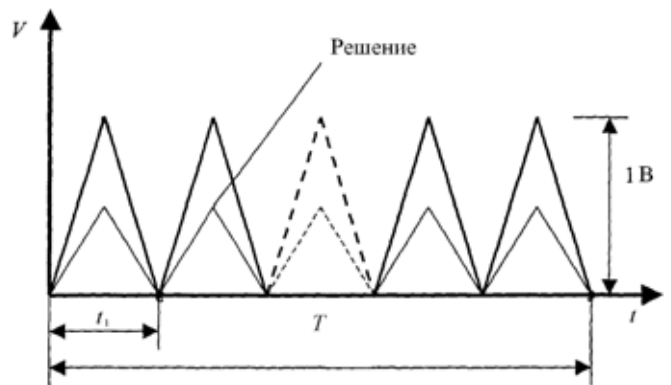


Рис. 9. Входной сигнал на источнике напряжения $V(t)$ и аналитическое решение

Необходимо получить численное решение для напряжения на емкости $C2$, которое должно совпадать с аналитическим решением, показанным тонкими линиями на рис. 9.

Математической моделью этой тестовой задачи будет система ДАУ вида (1) из трех уравнений:

$$\begin{cases} C1(dU_{C1}(t)/dt) - i(t) = 0; \\ C1(0,5 - U_{C2}(t)) \cdot (dU_{C2}(t)/dt) - i(t) = 0; \\ U_{C1}(t) + U_{C2}(t) - V(t) = 0 \end{cases} \quad (4)$$

относительно трех переменных $U_{C1}(t)$, $U_{C2}(t)$, $i(t) = i_{C1}(t) = i_{C2}(t)$. Аналитическое решение этой задачи (в модели все переменные рассматриваются безразмерными) для напряжения на емкости $C2$ при начальных условиях $U_{C1}(0) = 0$ и $U_{C2}(0) = 0$ имеет вид

$$U_{C2}(t) = C1 + C2(0) - \sqrt{(C1 + C2(0))^2 - 2C1V(t)}.$$

На рис. 10 приведен скриншот графика численного решения напряжения на емкости $C2$ при интегрировании системы ДАУ (4) для случая классической реализации неявного метода трапе-

ций, которое полностью совпало с аналитическим решением.

На рис. 11 приведен скриншот графика зависимости тока $i_{C2}(t)$, который значительно отличается от аналитического решения:

$$i_{C2}(t) = \frac{C1(0,5 - U_{C2}(t))}{\sqrt{(C1 + C2(0))^2 - 2 C1 V(t)}}.$$

Этот пример наглядно демонстрирует проблему "ложных колебаний" для неявного метода тра-

пеций и других АЛ-устойчивых неявных методов 4-, 6-, 8-, 10-го и т. д. порядков точности. Как отмечено выше, основной причиной появления "ложных колебаний" является наличие разрывов производных по времени в интегрируемых функциях (в данном примере это функция $V(t)$). Моменты времени разрыва производных либо известны (в данном случае через каждые 1 с), либо их можно определить, поэтому первоначально была сделана попытка разбить отрезок интегрирования на

отдельные "подотрезки" между моментами разрыва производных и интегрировать дифференциальные уравнения на этих "подотрезках" с последовательной передачей данных между "подотрезками". В пределах "подотрезка" гарантируется непрерывность всех интегрируемых функций и их производных, однако, даже начиная интегрирование на каждом "подотрезке" с минимальным шагом интегрирования, не удалось решить проблему "ложных колебаний".

В результате большого количества численных экспериментов выяснилось, что фактической причиной появления "ложных колебаний" является возможный идеальный скачок производных дифференцируемых и алгебраических переменных в конце шага численного интегрирования h_n (даже при $h_n \rightarrow 0$), т. е. возможный скачок значений отдельных элементов вектора XP и вектора Y , вызываемый разрывом производных у интегрируемых функций в начале (как в рассматриваемом примере) или в пределах текущего шага численного интегрирования h_n . Для неявного метода трапеций величина этих скачков значительно отличается от точных значений (даже при $h_n \rightarrow 0$), приводя к "ложным колебаниям". В то же время точное значение этих скачков теоретически можно получить в конце шага интегрирования h_n с помощью неявного метода Эйлера при $h \rightarrow 0$, так как при $h \rightarrow 0$ выполнение такого шага будет эквивалентно решению в конце

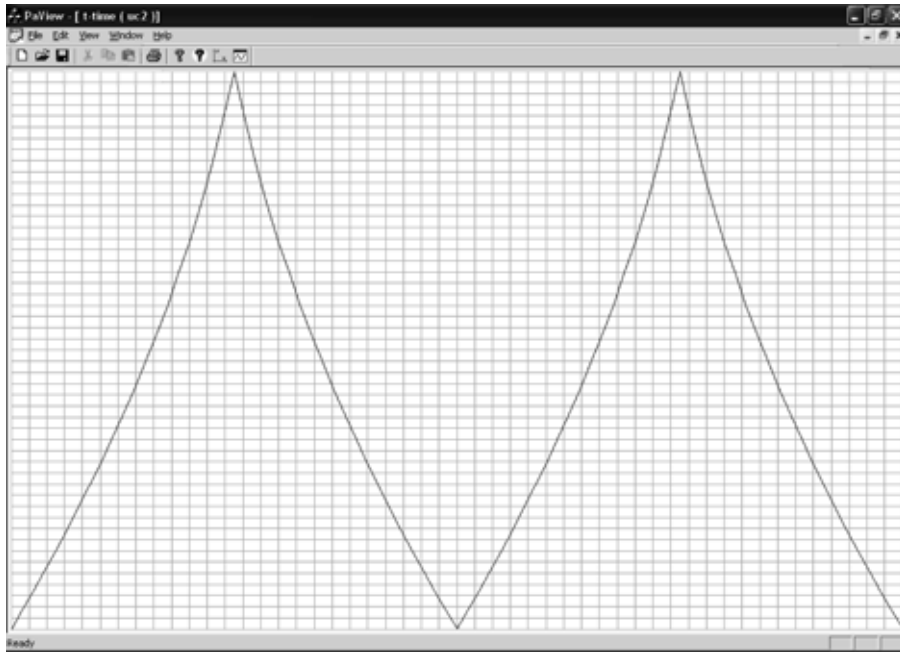


Рис. 10. Скриншот для численного решения $U_{C2}(t)$

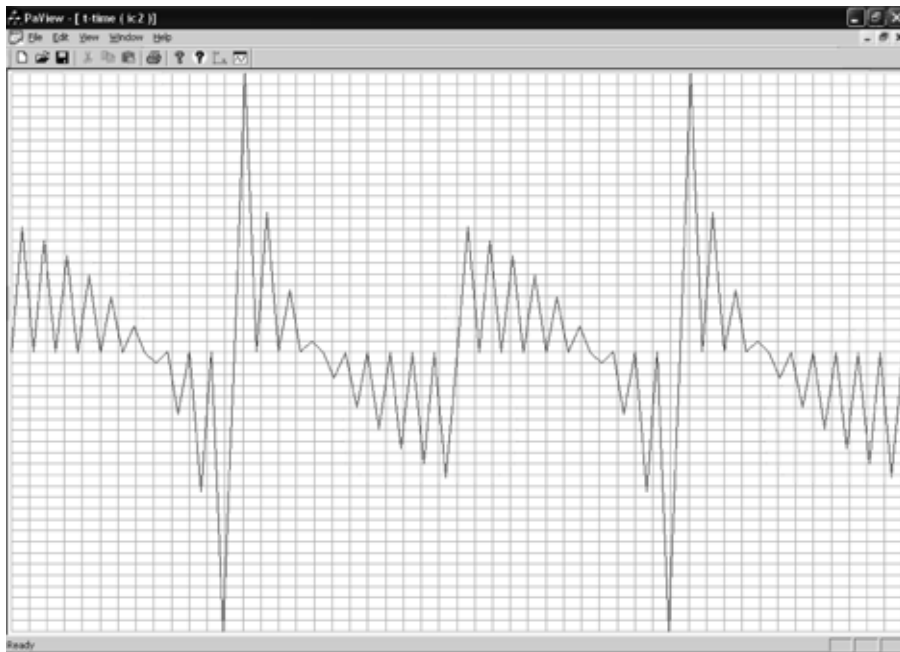


Рис. 11. Скриншот для численного решения $i_{C2}(t)$ неявным методом трапеций

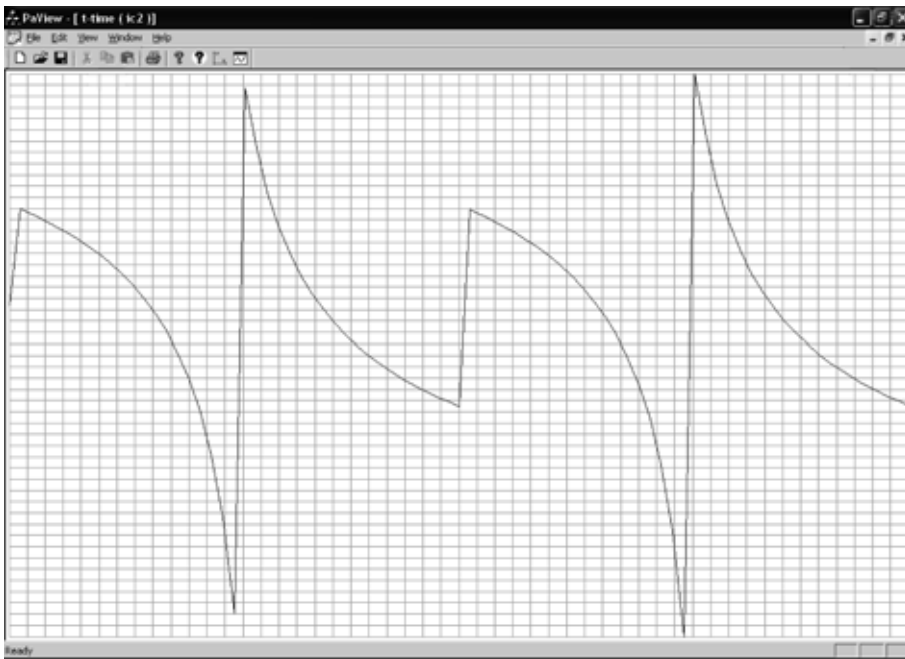


Рис. 12. Скриншот численного решения для тока $i_{C2}(t)$, полученного с помощью комбинированного корректирующего алгоритма

шага интегрирования системы нелинейных алгебраических уравнений $F(X, XP, Y, t) = 0$ ($m + k$)-го порядка относительно векторов XP и Y при фиксированном значении вектора X .

Использование корректирующего шага $h \rightarrow 0$ по неявному методу Эйлера при наличии разрыва производных дифференцируемых функций на соответствующем шаге интегрирования решило проблему "ложных колебаний" и позволило разработать комбинированный корректирующий алгоритм реализации неявного метода трапеций.

Комбинированный корректирующий алгоритм состоит из нулевого шага h_0 по неявному методу Эйлера при заданных начальных условиях $X_0 = X(0)$ аналогично комбинированному циклическому алгоритму и следующих основных этапов.

Этапы 1 и 2 аналогичны соответствующим этапам комбинированного циклического алгоритма.

Этап 3. Если в начале или в пределах текущего шага h_n присутствуют моменты времени, соответствующие разрывам производных интегрируемых функций, то выполняется коррекция значений векторов XP_n и Y_n . Для этого выполняется корректирующий шаг интегрирования h_{nadj} , много меньший шага h_n , по неявной формуле Эйлера путем решения системы нелинейных алгебраических уравнений

$$\begin{cases} XP_{nadj} = (X_{nadj} - X_n)/h_{nadj}; \\ F(X_{nadj}, XP_{nadj}, Y_{nadj}, (t_n + h_{nadj})) = 0 \end{cases}$$

и вычисления векторов X_{nadj} , XP_{nadj} и Y_{nadj} , затем вычисляются скорректированные значения векторов XP_n и Y_n (вектор X_n не изменяется!):

$$XP_n = XP_{nadj}; Y_n = Y_{nadj}.$$

На рис. 12 приведен скриншот графика численного решения для тока на емкости $C2$ для рассматриваемой задачи при интегрировании системы ДАУ (4) неявным методом трапеций, который был реализован с использованием комбинированного корректирующего алгоритма. Это решение полностью совпало с аналитическим решением.

Особенностью всех разностных формул неявного интегрирования ОДУ является деление на шаг интегрирования h . В работе [2] показано, что если про-

граммировать эти формулы в классическом виде, взятом из учебников, то шаг не должен быть меньше машинной точности вычислений, равной 10^{-8} для одинарной и 10^{-16} для удвоенной точности вычислений. При меньших значениях шага h программы интегрирования, реализующие неявные методы интегрирования, могут выдавать качественно неверное решение, что отмечено в работе [10]. В работе [5] предложен метод решения этой проблемы, а в дальнейшем была разработана программа DMAN, реализующая этот метод, а также корректирующий алгоритм для АЛ-устойчивых методов 2- и 4-го порядков точности. Для тестирования того, как в программах, реализующих неявные методы интегрирования, решена эта проблема, надо выполнять согласованное изменение параметров систем ОДУ таким образом, чтобы переменные и шаги интегрирования изменялись в максимально возможных пределах разрядной сетки процессора при сохранении качественно корректных результатов интегрирования. Принципы согласованного изменения параметров системы ОДУ изложены в [5] и выполнялись на примере расчета линейной электрической схемы с известным аналитическим решением (RLC: high Q filter — рис. 1 в [5]). В таблице приведены результаты тестирования по данному методу программы DMAN и наиболее известных программ, в которых реализованы неявные методы интегрирования: Radau-MathCAD (версия 2007 г.), odel5s-MATLAB (версия 2007 г.) и Rosenbrock-Maple (версия 2007 г.). В этой таблице приведены полученные диапазоны значений мас-

Масштабные коэффициенты K_i , K_u , K_t

Решатели ОДУ	Radau-MathCAD	ode15s-MATLAB	Rosenbrock-Maple	DMAN
$K_i = 1, K_u = 1$	$10^{-2} < K_t < 10^{250}$	$10^{-80} < K_t < 10^{250}$	$10^{-103} < K_t < 10^{250}$	$10^{-250} < K_t < 10^{250}$
$K_t = 1, K_u = 1$	$10^{-250} < K_i < 10^{250}$	$10^{-250} < K_i < 10^2$	$10^{-250} < K_i < 10^{250}$	$10^{-250} < K_i < 10^{250}$
$K_t = 1, K_i = 1$	$10^{-2} < K_u < 10^{250}$	$10^{-1} < K_u < 10^7$	$10^{-250} < K_u < 10^{250}$	$10^{-250} < K_u < 10^{250}$

штабных коэффициентов, предложенных в [5], для которых результаты расчета этой схемы еще совпадают с эталонным аналитическим решением (порядок масштабных коэффициентов при тестировании не задавался более 250 по абсолютной величине). За пределами приведенных значений программы MathCAD, MATLAB и Maple часто выдают качественно неверный результат без каких-либо сообщений.

В таблице приведены диапазоны значений масштабных коэффициентов по времени, току и напряжению (K_t , K_i , K_u), для которых результаты расчета тестовой задачи не совпадают с эталонным, аналитическим решением.

Продолжение статьи в следующем номере журнала.

Список литературы

1. Системы автоматизированного проектирования: Учеб. пособие для вузов: в 9 кн. / Под ред. И. П. Норенкова. М.: Высшая школа, 1986.
2. Жук Д. М., Маничев В. Б., Папсуев А. Ю. Обобщенный метод моделирования динамики технических систем // Информационные технологии. 2004. № 8. С. 6—14.
3. Норенков И. П., Евстифеев Ю. А., Маничев В. Б. Адаптивный метод ускоренного анализа многопериодных электронных схем // Изв. вузов. Радиоэлектроника. 1987. № 6. С. 35—39.

4. Маничев В. Б., Уваров М. Ю. Базовые методы интегрирования обыкновенных дифференциальных уравнений для программ анализа радиоэлектронных схем // Изв. вузов. Радиоэлектроника. 1989. Т. 32. № 6. С. 45—49.

5. Маничев В. Б. Новые алгоритмы для программ анализа динамики технических систем // Вестник МГТУ, сер. Приборостроение. 1996. Вып. 1. С. 48—56.

6. Деккер К., Вервер Я. Устойчивость методов Рунге—Кутты для жестких нелинейных дифференциальных уравнений. М.: Мир, 1988. 334 с.

7. Хайпер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи: Пер. с англ. М.: Мир, 1990. 512 с.

8. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие задачи: Пер. с англ. М.: Мир, 2001. 498 с.

9. Kundert K. The designer's guide to SPICE and Spectre. Kluwer Academic Publisher. 1995. 371 с.

10. Скворцов Л. М. Явный многошаговый метод численного решения жестких дифференциальных уравнений // Журнал вычислительной математики и математической физики. 2007. Т. 47. № 6. С. 959—967.

11. Норенков И. П., Жук Д. М., Маничев В. Б., Трудоношин В. А. Анализ электронных схем при совместном применении явных и неявных методов интегрирования // Изв. вузов. Радиоэлектроника. 1979. Т. 22. № 6. С. 25—28.

12. Евстифеев Ю. А., Маничев В. Б. Эффективный А-устойчивый метод интегрирования обыкновенных дифференциальных уравнений для программ анализа электронных схем // Изв. вузов. Радиоэлектроника. 1986. Т. 29. № 11. С. 31—35.

13. Маничев В. Б. Метод тестирования программ, реализующих неявные методы интегрирования ОДУ // Тез. докл. междунар. конф. "Современные проблемы вычислительной математики и математической физики". Самара. 2009. М.: МГУ им. М. В. Ломоносова, июнь 2009. С. 79.

УДК 004.942/629.3.017

Е. А. Перепелкин, д-р техн. наук, проф., e-mail: eap@list.ru,

В. И. Поддубный, канд. техн. наук, доц., e-mail: poddubny@list.ru,

Алтайский государственный технический университет им. И. И. Ползунова, г. Барнаул,

А. Варкентин, доктор, М. Ган, доктор,

фирма iXtronics GmbH, г. Падерборн, Германия, e-mail: info@iXtronics.com

Применение прикладного пакета SAMeL-View для моделирования управляемого движения колесного трактора

Приведено краткое описание основных объектов прикладного пакета SAMeL-View (Германия) для создания механических моделей механических систем и его возможностей. Рассмотрена модель колесного трактора с шарнирной рамой, созданная с использованием SAMeL-View. Приведены алгоритм синтеза ПИ-регулятора с наблюдателем в контуре управления и разработанная на его основе система управления движением колесного трактора. Приведенные результаты моделирования переходных процессов в MatLab и задаваемого движения трактора в SAMeL-View подтверждают адекватность его модели и системы управления движением.

Ключевые слова: управление, математическая модель, колесный трактор, пространство состояний, ПИ-регулятор

Введение

Одной из наиболее актуальных задач, решение которой в значительной мере способствует увеличению эффективности использования колесных машин, является повышение устойчивости движения и управляемости. Перспективным направлением, способствующим решению этой задачи, является программно-задаваемое управление колесными машинами при выполнении рабочих операций с использованием GPS-навигации.

Одним из этапов создания системы управления является разработка ее математической модели и математическое описание с использованием теории автоматического управления. Для разработки управления в инженерной практике применяют пакеты расширения MatLab (Simulink, Control System Toolbox, NSD Blockset). Их использование предполагает наличие математических моделей колесных машин в матричном виде. Следует отметить, что возможно получение системы дифференциальных уравнений, описывающих поведение колесной машины, традиционными методами с использованием уравнений Лагранжа 2-го рода на основании расчетной схемы. Однако для случая системы с несколькими степенями свободы (например, рассмотрение движения шарнирно-сочлененной машины в пространстве) эта задача существенно усложняется и увеличивается вероятность ошибки в аналитических расчетах. Кроме того, желательна проверка эффективности разработанного управления при моделировании рабочего движения колесной машины. При этом математическая модель колесной машины должна быть адекватна, т. е. должно быть учтено взаимодействие отдельных составных частей машины друг с другом и с опорной поверхностью. При выполнении этих требований происходит появление дополнительных степеней свободы и усложнение модели. Развитие вычислительной техники и современного прикладного объектно-ориентированного программного обеспечения позволяет провести разработку модели колесной машины, как мехатронной системы, состоящей из отдельных частей. Такие возможности предоставляют специализированные пакеты MatLab-SimMechanik, CAMeL-View и другие.

Разработка моделей механических систем в CAMeL-View

В прикладном пакете CAMeL-View, разработанном специалистами фирмы iXtronics GmbH (Падерборн, Германия), использована идеология визуального объектно-ориентированного программирования. Пакет предназначен для создания моделей различных механических систем, их анализа и оптимизации [1]. Для составления мо-

делей в пакете используются собственная библиотека объектов, позволяющая описывать отдельные тела системы, их механические связи между собой, выполнять математическое описание их взаимодействия. Моделируемая механическая система представлена как совокупность отдельных тел RigidBody, соединяемых с помощью сочленений Joint.

Для тела имеется возможность задавать его массово-геометрические характеристики (массу, тензор инерции); в качестве входа могут быть применены силы и моменты, приложенные в центре масс тела или портах (точках тела). Положение портов задается переменными, описывающими их положение относительно центра масс тела и поворот координатной системы порта относительно системы отсчета, жестко связанной с центром масс тела. Выходами для тела являются его кинематические параметры — линейное ускорение в инерциальной системе отсчета, линейные и угловые скорости в различных системах отсчета, текущие координаты.

Сочленение Joint определяет тип связи между телами. Его можно представить в общем случае, как пространственную пружину с задаваемыми жесткостью и демпфированием в направлениях осей X , Y , Z и угловыми жесткостью и демпфированием при угловых деформациях относительно осей. Сочленение может быть двух типов — кинематическое и динамическое. При соединении тел кинематическим сочленением Joint есть возможность обеспечить свободное (линейное или угловое) перемещение тел в желаемых направлениях. Таким образом, можно описать любой тип связи между телами (сферический или цилиндрический шарниры в пространстве, на плоскости и т. д.). Силы взаимодействия в высвобождающемся направлении для кинематического сочленения Joint описывают с помощью математического блока StateSpace. Обычно при описании сил используются кинематические выходы тела и задаваемые параметры. Если применяется динамическое сочленение Joint, то генерация сил упругого взаимодействия происходит автоматически с использованием задаваемых в блоке коэффициентов жесткости и демпфирования.

Математический блок StateSpace позволяет выполнять математические вычисления, применять программирование с использованием объектного языка программирования Objective-DSS, основой которого является язык Smalltalk. При программировании и вычислениях используют задаваемые в блоке параметры и его входы. В качестве входов могут быть скалярные, векторные величины, матрицы. При составлении топологической схемы модели обычно выходы твердого тела соединяют с входами StateSpace, имеющими

одинаковую размерность (становятся ее входами). Таким образом, кинематические параметры тела могут быть использованы при вычислениях в StateSpace. Выходами блока являются векторные, скалярные величины, матрицы. Довольно часто при разработке мехатронных моделей выходами StateSpace являются векторы сил и моментов взаимодействия тел системы, которые подаются на входы тел.

Перечисленные выше блоки являются наиболее важными для составления моделей (не умаляя значимости остальных библиотечных модулей). В состав библиотеки объектов входят также блоки таблиц, источники сигналов различной формы, цифровые фильтры, блоки регуляторов, передаточных функций и т. д. Применение библиотеки объектов позволяет создавать модели механических систем различной сложности с использованием элементов управления и регулирования.

Встроенное в CAMeL-View символическое ядро позволяет по составленной из библиотечных модулей топологической схеме модели генерировать ее математическую модель, представляющую собой систему обыкновенных дифференциальных уравнений (ОДУ) первого порядка в форме Коши. Пакет включает в себя также элементы линейного анализа — определение собственных значений матрицы управления, получение фазовой и амплитудно-частотной характеристик системы, линеаризация модели в заданной окрестности. Имеется возможность экспорта модели в MatLab-Simulink в виде модельного файла для дальнейшего его использования при разработке системы управления и генерации си-кода для использования модели при лабораторных испытаниях в режиме реального времени.

Для получения математической модели колесного трактора с шарнирной рамой в CAMeL-View бы-

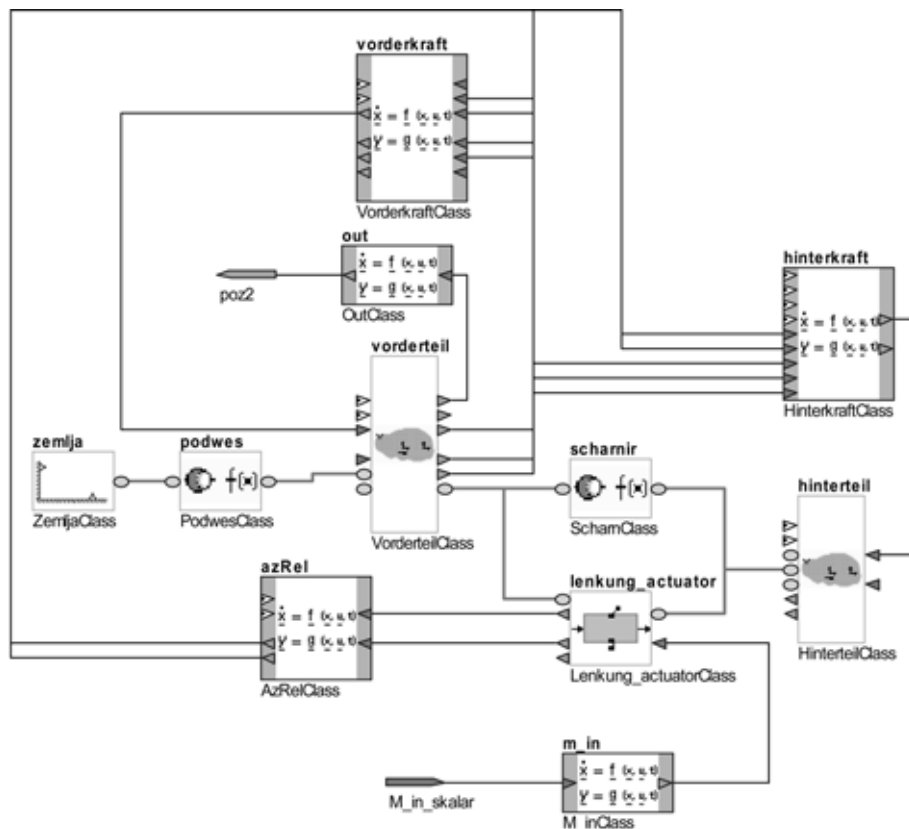


Рис. 1. Модель колесного трактора с шарнирной рамой

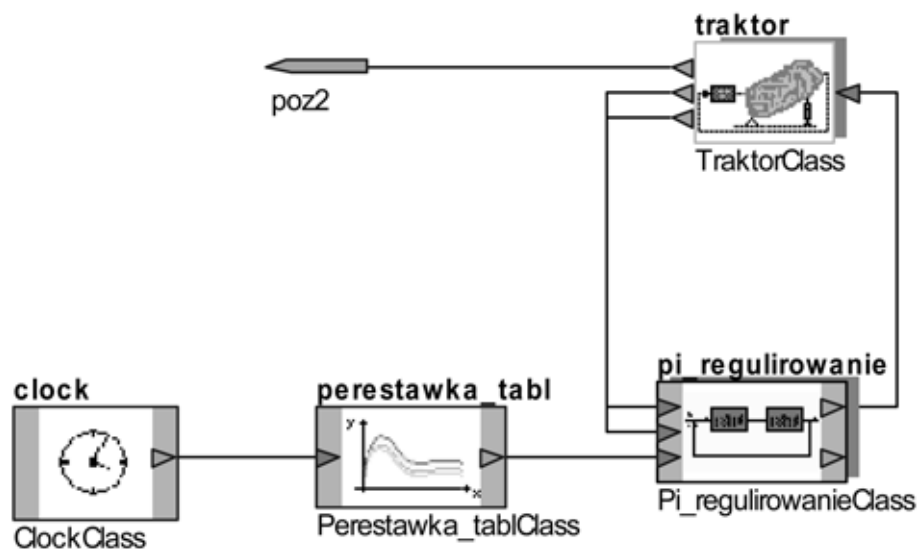


Рис. 2. Модель управляемого движения колесного трактора

ла разработана мехатронная модель, топологическое отображение которой представлено на рис. 1.

Передняя (**vorderteil**) и задняя (**hinterteil**) части трактора с помощью блока RigidBody представлены как твердые тела, соединенные кинематическим сочленением **scharnir**, допускающим взаимный поворот частей трактора. Трактор "вывешивается" относительно земли с помощью кинема-

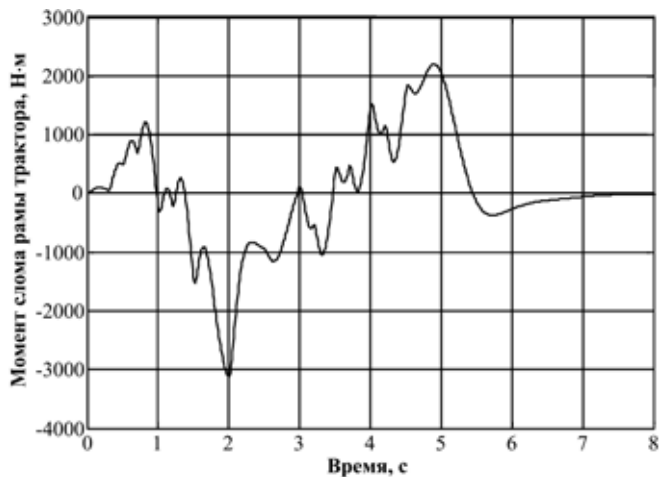


Рис. 3. Зависимость управляющего момента слома рамы трактора от времени

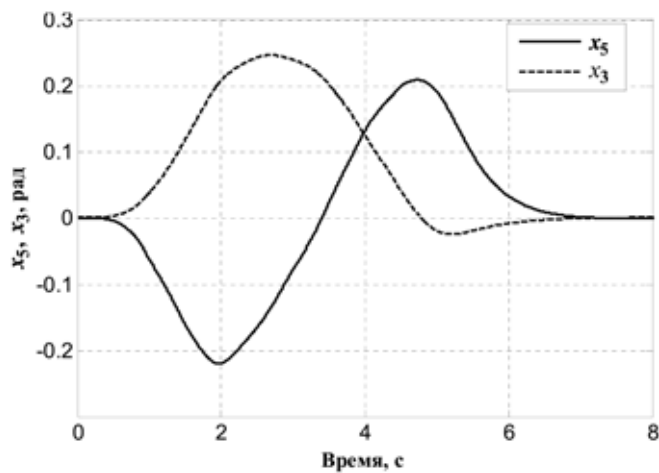


Рис. 4. Зависимость угла поворота продольной оси трактора x_3 и угла слома рамы x_5 от времени

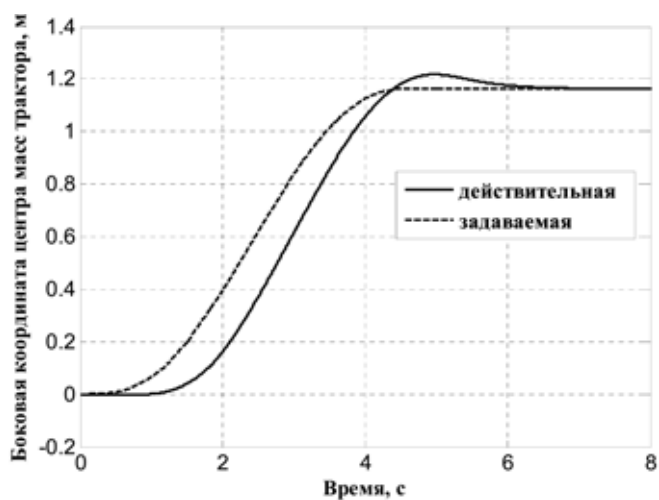


Рис. 5. Зависимость задаваемой и действительной боковой координаты центра масс трактора от времени

тического сочленения Joint **podwes**, допускающего плоское движение трактора в горизонтальной плоскости. Математическое описание сил, действующих на колеса трактора со стороны опорной поверхности, проводится с помощью блоков StateSpace — **vorderkraft** (переднее колесо) и **hinterkraft** (заднее колесо). Силы при моделировании определяются с использованием выходных текущих кинематических параметров звеньев трактора — линейных скоростей центров масс колес, угловых скоростей передней и задней частей трактора. Боковые силы определяются как функции угла бокового увода с использованием результатов экспериментальных данных [2]. Актуатор **lenkung_aktuator** вводит управляющий момент слома в сочленении между передней и задней частями трактора и определяет значение текущего угла слома рамы трактора. Выход блока (**out**) представляет собой боковую координату центра масс передней части трактора и является регулируемым выходом модели. Вход блока (**m_in**) — момент слома в шарнире трактора, является входом модели. С помощью пакета анализа была проведена линеаризация модели и определены матрицы для описания объекта в пространстве состояний.

Моделирование управляемого движения колесного трактора в CAMEL-View

Моделирование системы управления трактором с использованием ПИ-регулятора в контуре управления выполнялось для линеаризованной модели в среде MatLab, для полной нелинейной модели — в среде CAMEL-View. Алгоритм синтеза ПИ-регулятора приведен в прилож. 1, линеаризованная модель трактора в пространстве состояний в прилож. 2.

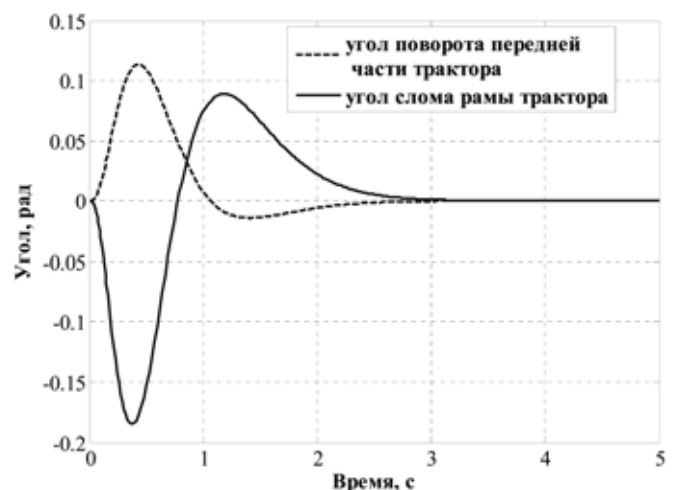


Рис. 6. Зависимость углов поворота передней части трактора и слома рамы трактора от времени

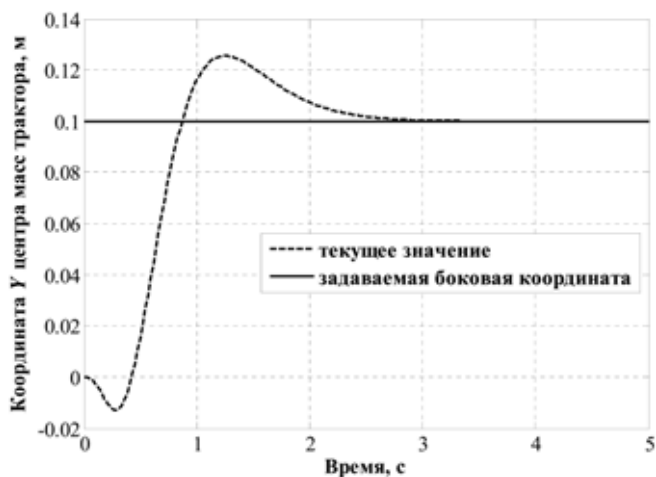


Рис. 7. Зависимость боковой координаты центра масс Y от времени

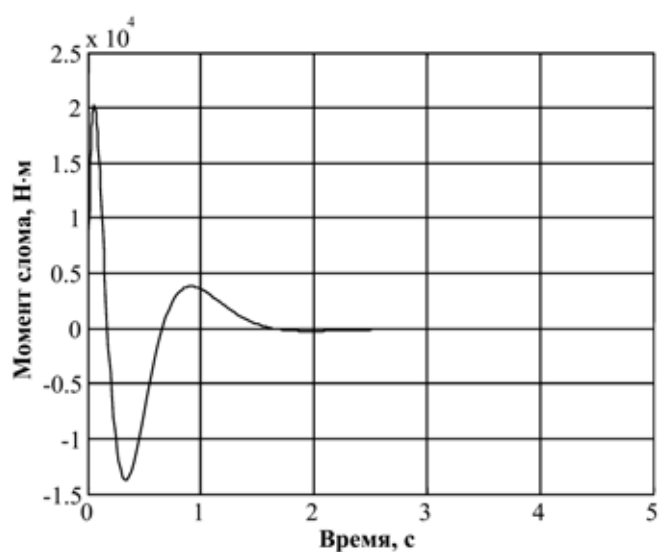


Рис. 8. Зависимость момента слома рамы трактора от времени

На рис. 2 изображена модель управляемого движения колесного трактора в САМeL-View. Для создания подсистем, содержащих механические объекты (твердые тела, сочленения, актуаторы), используется метакласс `MechtronicHcsOdss`, в случае подсистемы с блоками математических описаний — `MathematicHcsOdss`. За основу была взята полная нелинейная модель трактора, изображенная на рис. 1. В модели управляемого движения (рис. 2) она представлена как подсистема `traktor` (`MechtronicHcsOdss`), ПИ-регулирование с наблюдением реализовано в подсистеме `pi_regulowanie` (`MathematicHcsOdss`).

Для задания траектории движения центра масс трактора используется табличный блок `perestawka_tabl`, на вход которого подается текущее время, проставляемое блоком `clock`. Моделировали управление процессом смены полосы движения — переставка. На рис. 3–5 представлены результаты моделирования.

Результаты моделирования переходных процессов для линейной модели трактора изображены на рис. 6–8. Моделирование выполнялось при постоянной скорости продольного перемещения центра масс $x_2 = 2$ м/с и нулевых начальных значениях остальных переменных состояния. Регулируемыми переменными являются продольная x_1 и боковая x_3 координаты центра масс (прилож. 2). Вектор командного сигнала имеет вид:

$$\mathbf{v} = \begin{bmatrix} x_2 t \\ 0,1 \end{bmatrix}$$

где $x_2 t$ — командный сигнал по x_1 ; 0,1 — командный сигнал по x_3 .

Заключение

Качественные результаты моделирования управляемого движения трактора (закон изменения угла слома, траектории движения) показывают адекватность модели трактора реальному и работоспособность разработанной модели управления с использованием ПИ-регулирования. Дальнейшее развитие исследований будет направлено на практическую реализацию результатов в реальных системах управления рабочим движением трактора при выполнении полевых работ с использованием GPS-навигации для наблюдения.

Список литературы

1. **Hahn M.** Ein Objektmodell fuer den Mechatronikentwurf. Anwendung in der objektorientierten Modellbildung mechatronischer Systeme unter Verwendung von Mehrkorpersystemformalismen.: Dissertation zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.) dem Fachbereich 10 Maschinentechnik der Universitat-Gesamthochschule Paderborn. — Paderborn, 1999.
2. **Поддубный В. И., Павлюк А. С.** Влияние эксплуатационных параметров на характеристики увода пневматического колеса // Проблемы шин и резинокордных композитов. Математические методы в механике, конструировании и технологии: сб. научн. тр. М.: Изд. НИИШП, 1995. С. 42–46.
3. **Афанасьев В. Н., Колмановский В. Б., Носов В. Р.** Математическая теория конструирования систем управления. М.: Высш. шк., 2003.
4. **Параев Ю. И., Перепелкин Е. А.** Линейные матричные уравнения в задачах анализа и синтеза многосвязных динамических систем. Барнаул: Изд-во АлтГТУ, 2000.

Приложение 1

Алгоритм синтеза ПИ-регулятора

Объект управления описывается уравнениями

$$\begin{aligned}\dot{x} &= Ax + B_1u + B_2w, \\ y_1 &= C_1x; \\ y_2 &= C_2x,\end{aligned}$$

где x — n -мерный вектор состояния; u — m -мерный вектор управления; w — r -мерный вектор внешних неконтролируемых возмущений; y_1 — l -мерный вектор измерений; y_2 — p -мерный вектор регулируемых переменных. Необходимо построить управление в виде обратной связи по вектору измерений, при котором замкнутая система является асимптотически устойчивой и $\lim_{t \rightarrow \infty} y_1(t) = v$, где v — командный сигнал.

Будем считать, что возмущение и командный сигнал являются постоянными или медленно меняющимися функциями времени. Управление будем строить в виде ПИ-регулятора с наблюдателем в контуре управления [3]:

$$\begin{aligned}u &= F_1\bar{x} + F_2z; \\ \dot{\bar{x}} &= A\bar{x} + B_1u - L(y_1 - C_1\bar{x}); \\ \dot{z} &= y_2 - v,\end{aligned}$$

где \bar{x} — оценка состояния; F_1, F_2, L — матрицы обратной связи.

Замкнутая система описывается уравнениями

$$\begin{aligned}\dot{x} &= Ax + B_1F_1\bar{x} + B_1F_2z + B_2w; \\ \dot{\bar{x}} &= (A + B_1F_1 + L_1C_1)\bar{x} + B_1F_2z - LC_1x; \\ \dot{z} &= C_2x - v.\end{aligned}$$

Матрица замкнутой системы равна

$$\Phi = \begin{bmatrix} A & B_1F_1 & B_1F_2 \\ -LC_1 & A + B_1F_1 + LC_1 & B_1F_2 \\ C_2 & 0 & 0 \end{bmatrix}.$$

Если все собственные числа матрицы Φ имеют отрицательные действительные части, то замкнутая система асимптотически устойчива. Из асимптотической устойчивости следует, что при любых начальных условиях и при любых постоянных значениях w и v $\lim_{t \rightarrow \infty} y_1(t) = v$.

Рассмотрим задачу синтеза ПИ-регулятора как задачу о назначении желаемых собственных чисел матрицы Φ . Справедливо следующее.

Утверждение 1. Все собственные числа матрицы Φ можно произвольно задать, выбирая матрицы обратной связи F_1, F_2, L , тогда и только тогда, когда пара (A, B_1) управляема, пара (A, C_1) наблюдаема, $\text{rank} \begin{bmatrix} A & B_1 \\ C_2 & 0 \end{bmatrix} = n + p$.

$$\text{rank} \begin{bmatrix} A & B_1 \\ C_2 & 0 \end{bmatrix} = n + p.$$

Доказательство утверждения 1 следует из хорошо известных результатов теории линейных систем автоматического управления [3]. Отметим, что управляемость пары (A, B_1) означает, что

$$\text{rank}[B_1 \ AB_1 \ \dots \ A^{n-1}B_1] = n,$$

наблюдаемость пары (A, C_1) —

$$\text{rank} \begin{bmatrix} C_1 \\ C_1A \\ \vdots \\ C_1A^{n-1} \end{bmatrix} = n.$$

Матрицы обратной связи определим, решая два матричных уравнения Сильвестра [4], соответственно для пар матриц

$$\left(\begin{bmatrix} A & 0 \\ C_2 & 0 \end{bmatrix}, \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \right), (A^T, C_1^T),$$

сдвигая собственные числа матрицы замкнутой системы на комплексной плоскости влево на значение $\mu > 0$.

Приложение 2

Линеаризованная модель трактора в пространстве состояний

Линеаризованная модель трактора с шарнирной рамой, состоящего из двух звеньев, соединенных шарниром, включает переменные состояния: x_1 — продольная координата центра масс, м; x_2 — скорость продольного перемещения, м/с; x_3 — боковая координата центра масс, м; x_4 — скорость бокового перемещения, м/с; x_5 — угол поворота передней части трактора, рад; x_6 — угловая скорость передней части трактора, рад/с; x_7 — угол слома рамы трактора, рад; x_8 — угловая скорость задней части трактора, рад/с.

Переменными управления являются:

u_1 — момент сил, приложенный к шарниру (момент слома рамы), Н · м;

u_2 — движущая сила, создаваемая двигателем, Н.

Измерению доступны переменные x_1, x_3, x_7 . Регулируемыми переменными являются x_1, x_3 . Внешнее возмущение действует на боковое перемещение. Матрицы системы равны:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -15,44 & 30,85 & -2,92 & 1,47 & 1,2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8,19 & 16,370 & -8,03 & -3,19 & -2,6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 28,08 & -56,17 & -53,05 & -39,31 & -31,37 \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} 0 & 0 \\ 7,4e-05 & 0 \\ 0 & 0 \\ 0 & -1,39e-05 \\ 0 & 0 \\ 0 & 0,000212 \\ 0 & 0 \\ 0 & -0,000496 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Коэффициенты обратной связи, соответствующие значению $\mu = 3$, равны

$$\mathbf{F}_1 = \begin{bmatrix} -1,01e+06 & -1,012e+05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4,34e+06 & -2,44e+05 & -5,58e+06 & -2,23e+05 & -1,27e+06 & -1,82e+04 \end{bmatrix};$$

$$\mathbf{F}_2 = \begin{bmatrix} -1,68e+06 & 0 \\ 0 & -9,68e+06 \end{bmatrix}; \quad \mathbf{L} = \begin{bmatrix} -10 & 0 & 0 \\ -25 & 0 & 0 \\ 0 & -13,848 & 3,892 \\ 0 & 10,672 & -21,783 \\ 0 & -16,063 & 12,198 \\ 0 & 28,655 & 10,706 \\ 0 & 1,148 & -16,151 \\ 0 & -18,583 & 107,770 \end{bmatrix}.$$

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.925.8

Ю. Н. Косников, д-р техн. наук, зав. каф.,
Пензенский государственный университет,
e-mail: kosnikov@gmail.com

Геометрическое моделирование и отображение местности по топографическим данным

Предлагается технология отображения местности по физической географической карте: получение растрового образа карты, восстановление топографических данных, создание поверхностной геометрической модели, тесселяция, затенение, текстурирование.

Ключевые слова: сканирование карты, фильтрация изображения, скалярное поле, поверхностная геометрическая модель, тесселяция, затенение

Введение

Решение многих прикладных задач требует представлять на экране компьютерного монитора изображение земной поверхности. В ряде случаев это изображение должно нести информацию о пространственном рельефе местности, но в то же время может не быть фотореалистичным. Например, визуальная оценка пространственного рельефа нужна для принятия решений о проектировании дорог, трубопроводов, мостов. Изображение охраняемой зоны и находящихся в ней объектов помогает оценить потенциальную угрозу. Подготовка водителей наземного, воздушного и водного транспорта на тренажерах первоначального обучения становится более эффективной, если сделать изображение визуальной обстановки не

условным, а "привязанным" к конкретной местности.

Разработчики и пользователи систем отображения визуальной обстановки знают, что 60 % и более их стоимости падает на визуальную базу данных. Это связано с высокой трудоемкостью создания модели реального рельефа. Например, существуют специализированные программы — генераторы ландшафтов, такие как Terragen, Genesis, Bryce. Они строят рельеф по карте высот, которую создает в интерактивном режиме оператор или импортирует из программ геометрического моделирования. В то же время существует общедоступная информация о рельефе земной поверхности: это физические географические карты и топографические планы местности. Их используют, например, в геоинформационных системах (ГИС), некоторые из которых имеют средства для отображения пространственного географического рельефа. При этом доля ручного труда оператора в них остается высокой [1].

В данной работе предлагается автоматизированная технология геометрического моделирования и отображения местности на основе топографических данных.

В процессе получения визуальной модели необходимо решить две задачи. Первая задача — восстановление пространственного (3D) рельефа по его 2,5D-представлению, т. е. по плоскому представлению, содержащему информацию о координате высоты в косвенной форме. Вторая задача — отображение участка местности в наглядном виде, а именно, с характерной раскраской и наложением теней.

1. Этапы процесса моделирования и отображения местности

Процесс геометрического моделирования и отображения местности по топографическим данным включает ряд этапов. Сначала исходные данные должны быть преобразованы в форму, пригодную для компьютерной обработки. Затем из 2,5D-представления необходимо получить математическую 3D-модель в виде скалярного поля. Она должна содержать описание высот в характерных точках рельефа, а также описание нормалей к поверхности в этих точках, что необходимо для дальнейшего затенения рельефа. Далее следует выбрать вид геометрического примитива и на его основе разработать аналитическую или кусочно-аналитическую модель. Следующим этапом является переход к полигональной форме модели. Именно полигональная модель передается в стандартный графический процессор компьютера для визуализации. Для получения полигональной модели криволинейный рельеф, описанный в кусоч-

но-аналитической форме, разбивается на фрагменты, т. е. подвергается тесселяции. Еще одним этапом получения изображения является изготовление текстуры, представляющей собой рисунок, характерный для выбранного участка местности. Наложение текстуры на полигональную модель, а также моделирование освещенности рельефа (затенение) осуществляются средствами графического процессора. Рассмотрим содержание перечисленных этапов.

2. Получение исходных данных для компьютерного моделирования

Выбор источника исходных данных определяется решаемой задачей. Источником информации о рельефе большой протяженности может являться физическая географическая карта. Детальный рельеф небольшой протяженности представляется планом местности или результатами непосредственных топографических замеров. В последнем случае рельеф уже представлен скалярным полем высот, а в случае использования карт он должен быть приведен к этой форме. Первым естественным шагом является обработка карты сканером цветного изображения и представление результатов в графическом формате BMP. Другим вариантом является отыскание картографической информации в сети Internet и вырезание нужного фрагмента карты средствами растрового графического редактора.

На физической карте информация о высоте рельефа представлена путем цветового кодирования. Однако цветовая информация о рельефе "затенена" ненужными символами надписей и условных обозначений. Они имеют другой цвет, в идеале — черный, а на практике представляющий собой различные градации серого и близких к нему цветов. Для удаления ненужных цветовых элементов следует применить алгоритмы фильтрации изображений [2]. Их основой является попиксельный анализ изображения карты и отыскание пикселей, цвет которых отсутствует в приведенной на физической карте шкале цветов. Цвет каждого такого пикселя заменяют усредненным цветом соседних пикселей.

Далее цветовую информацию необходимо использовать для восстановления пространственного рельефа местности. Однако в силу отклонений в составе красок и бумаги одни и те же высоты на картах в различных полиграфических изданиях представляются с помощью различных цветовых оттенков. Это различие не очень велико, но для приближения синтезированного рельефа к истинному виду оно требует усложнения компьютерной обрабатываемой программы. Различие цветов устраняется путем приведения текущих цветов

картографического изображения к принятой ("стандартной") палитре шкалы высот. Приведение идет попиксельно. Его критерием для j -го пикселя является минимум расстояния r_{i-j} между цветовым оттенком этого пикселя и i -м цветом стандартной шкалы высот в цветовом пространстве RGB :

$$r_{i-j}(C_{sj}) = \min_{i=1..N} \{(R_j - R_{si})^2 + (G_j - G_{si})^2 + (B_j - B_{si})^2\},$$

где C_{sj} — приведенный к стандартной шкале цвет j -го пикселя; R_j, G_j, B_j — значения цветовых компонентов j -го пикселя; R_{si}, G_{si}, B_{si} — значения цветовых компонентов i -го цвета шкалы высот; N — число цветов шкалы высот.

3. Создание математической модели участка местности

Цветояркость каждой точки изображения местности может быть поставлена в соответствие высоте рельефа в этой точке. Однако точечная модель избыточна и требует для хранения большого объема памяти. Для описания рельефа достаточно выделить из всего множества точек ограниченный набор характерных (опорных) точек, расположенных в узлах ортогональной координатной сетки. Для построения геометрической модели рельефа эти точки могут стать вершинами плоских полигонов. В силу того, что число цветов в шкале высот географических карт сравнительно невелико (от 7 до 18), невелико и число высот, которые могут быть использованы для геометрического моделирования. При восстановлении рельефа это приводит к наличию множества плоских горизонтальных участков разного уровня ("террас"). В итоге рельеф теряет наглядность. В связи с этим имеет смысл отказаться от полигональной модели рельефа и положить в ее основу геометрический примитив криволинейной формы.

Традиционными для графических систем криволинейными примитивами являются сплайновые отсеки. Однако в рассматриваемом случае они имеют ряд недостатков. Например, сплайн Безье описывается 16 опорными точками, из которых четыре являются угловыми точками отсека, а остальные задают его форму. Четверка узлов координатной сетки рельефа может быть принята за угловые точки отсека Безье, но остальные 12 точек сплайна требуют определения, причем никаких дополнительных данных, кроме исходного набора опорных точек, для этого нет. Другой весьма распространенный примитив — B -сплайн в общем случае не проходит ни через одну из своих опорных точек, что дает погрешность восстановления рельефа.

Причина перечисленных трудностей в том, что при изготовлении карт в результате дискретиза-

ции высот рельефа теряется информация о топологической тенденции его участков. Чтобы восстановить ее, хотя бы с некоторой погрешностью, предлагается анализировать направления нормалей к поверхности рельефа. Как известно, координаты нормального вектора $n_{xij}, n_{yij}, n_{zij}$, который проходит через некоторую точку P_{ij} поверхности, определяются значениями частных производных функции, описывающей эту поверхность (i, j — номера узла координатной сетки по осям x и y). Для явной формы описания поверхности $z = f(x, y)$ эти координаты равны [3]

$$n_{xij} = \left(\frac{\partial z}{\partial x} \right)_{ij}, \quad n_{yij} = \left(\frac{\partial z}{\partial y} \right)_{ij}, \quad n_{zij} = -1. \quad (1)$$

В рассматриваемом случае аналитическое описание поверхности отсутствует, его только предстоит создать. Однако в любом случае на топологию поверхности в районе опорной точки влияет расположение соседних опорных точек. В работе [4] показано, что значения частных производных функции в некоторой опорной точке могут быть приняты тангенсы углов наклона прямолинейных отрезков, соединяющих точки, соседние по отношению к выбранной точке в направлениях координатных осей x, y . Тогда координаты вектора нормали n_x, n_y , проходящего через опорную точку P_{ij} , можно найти из выражений

$$n_{xij} = \frac{z_{(i+1)j} - z_{(i-1)j}}{2R}, \quad n_{yij} = \frac{z_{i(j+1)} - z_{i(j-1)}}{2R}, \quad (2)$$

где R — шаг координатной сетки, а третья координата нормали принимает единичное значение. Полученную информацию о направлениях нормалей, проходящих через опорные точки карты высот, можно использовать для описания отсеков поверхности, т. е. геометрических примитивов. Например, можно найти недостающие опорные точки сплайна Безье, которые определяются именно углами наклона касательных, проведенных через угловые точки отсека.

Более простым, чем сплайны, но эффективным в изобразительном отношении является рельеф-примитив, или R -примитив [5, 6]. Это четырехугольный отсек криволинейной поверхности, который строго проходит через угловые точки и имеет форму, учитывающую топологическую тенденцию окружающего рельефа. Координаты текущей точки R -примитива находят как взвешенные суммы одноименных координат опорных точек. Форму отсека задают путем рационального выбора весовых (смешивающих) функций, значения которых зависят от расстояния \bar{k} между текущей точкой и опорными точками вдоль координатной оси k ($k = x, y$). Смешивающая функция R -примитива представляет собой степенной полином вида

$$BF_k = (1 - \bar{k})^{m_k}, \quad (3)$$

где степень m_k (характеристика рельефа) может превышать 3 и быть меньше 1. Это улучшает формообразующие возможности примитива. В то же время для снижения ресурсоемкости вычислений число разновидностей смешивающих функций ограничивается. Другими словами, в вычислительной системе создается алфавит смешивающих функций, значения которых могут быть вычислены заранее с определенным шагом.

Математическое описание R -примитива включает координаты четырех угловых опорных точек и их характеристики рельефа. Общий вид описания в декартовой системе координат следующий:

$$z = \sum_{i=1}^4 z_i BF_{xi} \cdot BF_{yi}, \quad (4)$$

$$BF_{ki} = \begin{cases} (1 - \bar{k})^{m_{ki}} & \text{при } \bar{k} < 1, \\ 0 & \text{при } \bar{k} \geq 1, \end{cases} \quad (5)$$

$$\bar{k}_i = \frac{|k - k_i|}{R}, \quad (6)$$

где z_i — высота i -й опорной точки; BF_{ki} — смешивающая функция i -й опорной точки, действующая вдоль координаты k ($k = x, y$); k_i — координата x, y i -й опорной точки; \bar{k}_i — относительное расстояние между текущей точкой поверхности примитива и i -й опорной точкой вдоль координаты k ; m_{ki} — характеристика рельефа i -й опорной точки, действующая вдоль координаты k ; R — шаг координатной сетки в плоскости аргументов.

В соответствии с выражениями (4–6) высота текущей точки примитива представляет собой сумму "вкладов", сделанных опорными точками. Значение вклада каждой опорной точки зависит от ее близости к текущей точке и определяется значением смешивающей функции этой опорной точки. Если текущая точка удалена от опорной точки на шаг R и более, то она не находится под влиянием этой опорной точки и не получает от нее вклада в координату высоты.

Алгоритм генерации R -примитива (рис. 1) состоит из следующих укрупненных шагов.

1. Примитив задается высотами (координатами z) четырех угловых опорных точек. На рис. 1 это точки P_1, P_2, P_3, P_4 . Каждая опорная точка получает две характеристики рельефа m_{xi}, m_{yi} ($i = 1, \dots, 4$), действующие вдоль координатных осей.

2. Организуется сканирование плоскости аргументов x, y с некоторым шагом.

3. Каждая очередная точка на плоскости аргументов (например, точка PP на рис. 1) принимается за проекцию текущей точки примитива (точ-

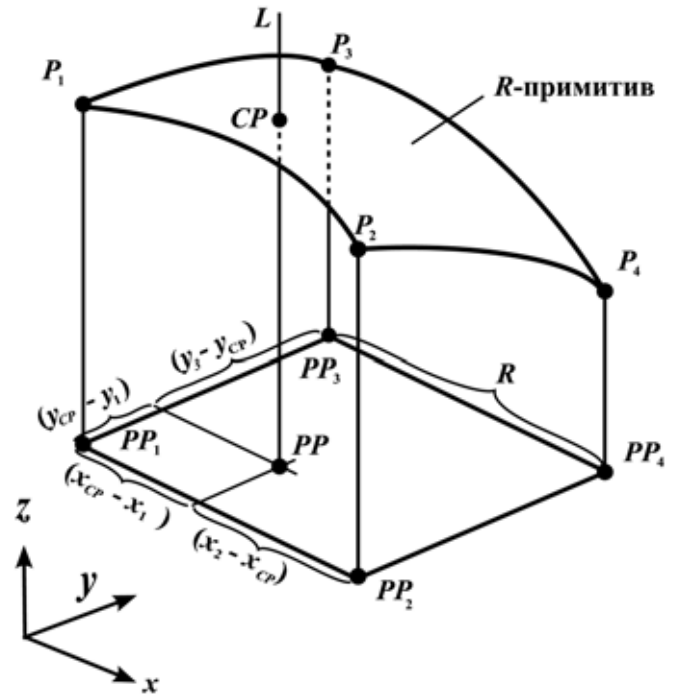


Рис. 1. Криволинейная поверхность R -примитива

ки CP). Положение (высоту) точки CP на проекторе L нужно найти.

4. Для определения положения точки CP найти приведенные к величине R расстояния \bar{x}_i, \bar{y}_i от проекции текущей точки (точки PP) до проекций опорных точек PP_1, PP_2, PP_3, PP_4 .

5. По выражениям (4–6) вычисляется высота текущей точки.

Каждая опорная точка в локальной системе координат R -примитива получает две характеристики рельефа m_{xi}, m_{yi} , которые определяют вид полинома (3) и форму примитива вблизи опорной точки. Если алфавит смешивающих функций содержит n их разновидностей, то число вариантов рельефа примитива составит n^8 , что позволяет получать самые разнообразные формы уже при малых n . Значения m_{xi}, m_{yi} для каждой опорной точки определяются топологией рельефа в окрестности этой опорной точки. Для их нахождения можно использовать информацию о направлениях нормалей в опорных точках, получение которой описано ранее (выражение (2)). Методика нахождения характеристик рельефа приведена в работе [7]. При переходе от одного примитива к следующему необходимо решать проблему их стыковки. В работах [5, 6] показано, что составная поверхность, получающаяся на основе рельеф-примитивов, не имеет разрывов и строго проходит через опорные точки.

4. Получение полигональной модели рельефа

Полигональная модель получается в результате тесселяции криволинейной поверхности. Суть тес-

селяции — это нахождение на поверхности каждого R -примитива расположенных по некоторому закону дополнительных точек. Соседние дополнительные точки объединяются и принимаются за вершины плоских, обычно треугольных, полигонов. Набор смежных полигонов образует полигональную сеть, а ее описание — полигональную модель рельефа. Координаты x и y вершин полигонов получают в процессе ортогонального сканирования плоскости аргументов R -примитива с заданным шагом. Размер этого шага определяет детальность полигональной сети. Высоты вершин вычисляются по выражениям (4)—(6). Параметры смешивающих функций BF_{xi} , BF_{yi} в выражении (4) представлены в относительных единицах. Это дает возможность вычислить их значения с различным шагом заранее и занести в память графической системы в виде таблицы. В процессе формирования полигональной сети значения BF_{xi} , BF_{yi} считываются из таблицы с минимальными затратами времени.

Важным фактором является закон обхода плоскости аргументов R -примитива в процессе тесселяции. Он должен учитывать особенности рисования полигонов средствами графической системы. Например, в широко применяемой графической библиотеке OpenGL существует команда рисования последовательности (стрипа) треугольных полигонов:

```
glBegin(GL_TRIANGLE_STRIP)
< Описание множества вершин >
glEnd();
```

Ее параметрами являются координаты вершин полигональной сети. Первые три вершины эта команда воспринимает как вершины первого треугольника стрипа. Каждая следующая вершина воспринимается как третья вершина следующего треугольника, а в качестве первых двух вершин этого следующего треугольника используются две последние вершины предыдущего треугольника [8]. Процесс тесселяции рельефа с применением названной команды организован в два этапа. На первом вычисляются и заносятся в память графической системы высоты вершин полигональной сети. На втором этапе эти высоты используются для описания (рисования) сети с помощью стрипов треугольников.

5. Моделирование освещения рельефа

В компьютерной графике эта задача непосредственно связана с геометрией, так как для ее решения требуется провести анализ пространственного положения осветителя и поверхности. Анализируется угол между вектором луча света L , падающего в некоторую точку поверхности, и вектором нормали, проведенным через эту точку [3]. В качестве таких точек поверхности используют вершины полигональной сети. Вектор нормали к полигону можно найти из векторного про-

изведения векторов, совпадающих с ребрами полигона [4]. Однако в случае использования R -примитивов более эффективен другой подход. Описание R -примитива — аналитическая функция, представленная в явной форме. Тогда, как уже отмечалось, координаты нормального вектора определяются выражениями (1).

Рассмотрим вычисление частной производной функции (4) по аргументу x с учетом (5) и (6):

$$\frac{\partial z}{\partial x} = \sum_{i=1}^4 z_i \cdot m_{xi} (1 - \bar{x}_i)^{m_{xi}-1} (1 - \bar{y}_i)^{m_{yi}} \frac{\partial \bar{x}}{\partial x}.$$

Для нахождения производной ($\partial \bar{x} / \partial x$) нужно принять во внимание, что \bar{x} является относительным расстоянием между текущей и i -й опорной точками. Расстояние — положительная величина, поэтому в выражение (6) входит функция абсолютного значения. Ее можно представить так:

$$\bar{x} = \frac{|x - x_i|}{R} = \frac{1}{R} (x - x_i) \operatorname{sgn}(x - x_i).$$

Тогда с учетом равенства $\frac{\partial (\operatorname{sgn}(x - x_i))}{\partial x} = 0$

$$\text{получаем } \frac{\partial \bar{x}}{\partial x} = \frac{1}{R} \operatorname{sgn}(x - x_i).$$

В итоге

$$\frac{\partial z}{\partial x} = \sum_{i=1}^4 z_i \frac{m_{xi}}{R} (1 - \bar{x}_i)^{m_{xi}-1} (1 - \bar{y}_i)^{m_{yi}} \operatorname{sgn}(x - x_i),$$

аналогично

$$\frac{\partial z}{\partial y} = \sum_{i=1}^4 z_i \frac{m_{yi}}{R} (1 - \bar{x}_i)^{m_{xi}} (1 - \bar{y}_i)^{m_{yi}-1} \operatorname{sgn}(y - y_i).$$

Введя очевидные обозначения, получим следующую форму записи последних выражений:

$$\frac{\partial z}{\partial x} = \sum_{i=1}^4 z_i \overline{BF}_{xi} \overline{BF}_{yi} \operatorname{sgn}(x - x_i); \quad (7)$$

$$\frac{\partial z}{\partial y} = \sum_{i=1}^4 z_i \overline{BF}_{xi} \overline{BF}_{yi} \operatorname{sgn}(y - y_i), \quad (8)$$

где \overline{BF}_{xi} , \overline{BF}_{yi} — смешивающие функции производных. Подобно BF_{xi} , BF_{yi} их значения можно рассчитать заранее и занести в память графической системы для последующих табличных вычислений.

Координаты нормалей вычисляются по выражениям (1), (7), (8) для всех вершин, за исключением опорных точек, где $\operatorname{sgn}(x - x_i) = 0$. Для опорных точек берут усредненные нормали, координаты которых найдены по выражениям (2). Тогда при переходе через опорную точку направления нормалей меняются плавно, чему также способствует применение модели освещения Гуро или Фонга, поддерживаемой графической системой [3]. Тем самым достигается правильное освещение рельефа.

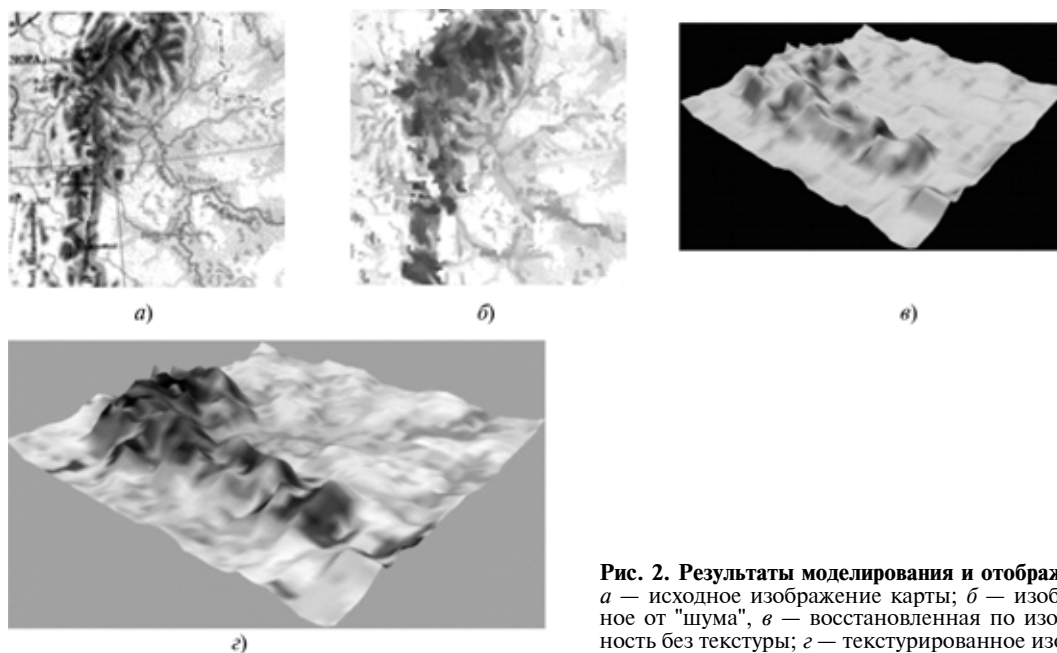


Рис. 2. Результаты моделирования и отображения местности: *a* — исходное изображение карты; *b* — изображение, очищенное от "шума"; *в* — восстановленная по изображению поверхность без текстуры; *г* — текстурированное изображение рельефа

6. Текстурирование рельефа

Если к изображению ландшафта не предъявляются высоких требований по реалистичности, то простым решением является использование в качестве текстуры рельефа самой исходной карты, представленной в графическом формате. В описываемой технологии предлагается изображение карты в формате BMP использовать для дальнейшего построения геометрической модели рельефа. Такой прием гарантирует, что полученные из этого изображения текстура и поверхность будут согласованы по размерам и детальности. Такая согласованность означает, что при наложении текстуры на рельеф ее не требуется ни сильно сжимать, что приводит к пропаданию мелких деталей, ни сильно растягивать, что приводит к пикселизации изображения.

Наложение текстуры осуществляют штатными средствами графической системы. Например, в графической библиотеке OpenGL для этого служат команды `glTexCoord`, `glTexCoordPointer` [8]. Они сопоставляют точки текстуры с опорными точками (вершинами) поверхности рельефа. Для правильного наложения текстуры законы обхода текстурных и геометрических (при тесселяции) координат должны быть идентичными.

Заключение

Предложенная технология геометрического моделирования и отображения местности по топографическим данным позволяет получать наглядные пространственные ландшафты с минимальным участием человека. Изобразительные возможности технологии демонстрирует рис. 2,

изображения которого получены студентами-магистрантами М. Зубковым и В. Саушевым. На рис. 2, *a* представлен взятый из сети Internet фрагмент физической географической карты. На рис. 2, *b* этот фрагмент показан после удаления координатной сетки и надписей. Полученное после фильтрации изображение в дальнейшем использовано в качестве текстуры рельефа. На рис. 2, *в* без наложения текстуры приведена поверхность, восстановленная по изображению карты. В качестве геометрического примитива использован рельеф-примитив. Наконец, на рис. 2, *г* представлено изображение рельефа после наложения текстуры. Рендеринг изображения выполнен средствами графической библиотеки OpenGL.

Список литературы

1. **Вишневская Е. А., Елобогоев А. В., Высоцкий Е. М., Добрецов Н. Н.** Пространственное моделирование рельефа средствами ГИС для морфотектонического анализа // *Материалы Международной конференции "Интеркарто-6"*. 22–24 августа 2000 г. Апатиты, 2000. С. 57.
2. **Павлидис Т.** Алгоритмы машинной графики и обработки изображений. М.: Радио и связь, 1986. 417 с.
3. **Иванов В. П., Батраков А. С.** Трехмерная компьютерная графика / Под ред. Г. М. Полищука. М.: Радио и связь, 1995. 224 с.
4. **Фокс А., Пратт М.** Вычислительная геометрия. Применение в проектировании и на производстве: Пер. с англ. М.: Мир, 1982. 304 с.
5. **Косников Ю. Н.** Компьютерное моделирование поверхностей свободных форм // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2005. № 5. С. 91.
6. **Косников Ю. Н.** Геометрическое моделирование в графических системах реального времени: монография. Пенза: Информац.-издат. центр Пенз. гос. ун-та, 2006. 218 с.
7. **Косников Ю. Н.** Методика геометрического моделирования поверхностей по карте высот // *Аэрокосмические приборные технологии: Сборник материалов Третьего международного симпозиума*. СПб, 2004. С. 213.
8. **OpenGL.** Официальный справочник: Пер. с англ. / Под ред. Д. Шрайнера. — СПб.: ДиаСофтЮП, 2002. 512 с.

В. А. Немтинов, д-р техн. наук, доц.,
А. А. Горелов, канд. техн. наук, доц.,
К. В. Немтинов, студент,
Д. В. Кулин, студент,

Тамбовский государственный технический
 университет, г. Тамбов,
 e-mail: e-nemtinov@mail.gaps.tstu.ru

Пространственно-временная модель объекта культурно- исторического наследия

Рассмотрена модель информационного объекта культурно-исторического значения, представленная в виде графовой структуры фреймов и включающая сведения о составе, свойствах объекта и его элементах, а также способах задания значений этих свойств, которая позволяет специалисту систематизировать всю информацию о реальном физическом объекте, упорядочить ее хранение на электронных носителях и обеспечить эффективную обработку.

Ключевые слова: объект культурно-исторического значения, информационный объект, аналитические и информационно-логические модели

Введение

В настоящее время наиболее полное исследование объектов культурного и исторического наследия возможно только на основе многостороннего подхода с применением различных информационных потоков, затрагивающих не только зрительное содержание. Решить такую задачу можно средствами современных компьютерных технологий. В этом случае геоинформационные системы служат главной цели — построению виртуального зрительного образа объекта. Последний, в свою очередь, дополняется разнообразной атрибутивной информацией, что приводит к построению единого информационного пространства территории и его пространственно-временной модели (если, например, речь идет о мемориальных военно-исторических комплексах, достопримечательных местах, ансамблях и т. п.).

Такой подход, как показывает наш опыт, позволяет построить экскурсионный характер исследования объекта. При этом исследователь может самостоятельно выбрать маршрут экскурсии и двигаться по нему, зрительно изучая предмет исследования. Многочисленная и разноплановая информация об объекте может быть в любой момент востребована им из информационно-аналитической базы данных. Такой вариант может быть

успешно использован в сети Интернет и послужить благой цели — изучению объектов исторического и культурного наследия, расположенных на территории нашей страны.

Практическая реализация такого подхода в настоящее время невозможна без использования прикладных автоматизированных информационных систем (АИС), которые оперируют знаниями предметной области. При этом одними из основных элементов АИС являются модели объектов, входящих в состав той или иной культурной исторической группы, и модели представления знаний об этих объектах.

Модель информационного объекта

Введем понятие информационного объекта для комплекса исторического и культурного наследия.

Информационным объектом комплекса исторического и культурного наследия (реального объекта, процесса или события) называется формализованная совокупность знаний о нем, представленная в виде графовой структуры фреймов и включающая сведения о составе, свойствах комплекса и его элементах, а также о способах задания значений и/или графического образа этих свойств.

Схема представления данных об информационном объекте O , описывающем комплекс исторического и культурного наследия, приведена на рис. 1:

$$O = \{St_o, S_o, Sp_o, M_o, M_o^g\}, \quad (1)$$

где St_o — фрейм, описывающий структурный состав физического объекта [1]; S_o — фрейм, описывающий свойства, характерные для всего объекта; Sp_o — множество способов задания свойств объекта; M_o — множество моделей, позволяющих определить значения свойств, характерных для всего объекта; M_o^g — множество моделей графических образов свойств, для которых он актуален (отдельные свойства могут не иметь графического образа).

При этом следует отметить, что

$$s_{oi} = \{s'_{oi}, z'_{oit}, g'_{oit}\}, \quad i = \overline{1, N}; t \in [0, T];$$

где s'_{oi} , z'_{oit} , g'_{oit} — соответственно наименование слота свойства s_{oi} , его значение и графический образ; N — число свойств; T — время жизненного цикла объекта.

Элементами множества Sp_o являются термы:

$Sp_o = \{\text{"задается ЛПР"}, \text{"выбирается ЛПР из списка"}, \text{"рассчитывается по модели"}, \text{"описывается лингвистической моделью"}, \text{"изображается графической моделью"}\}.$

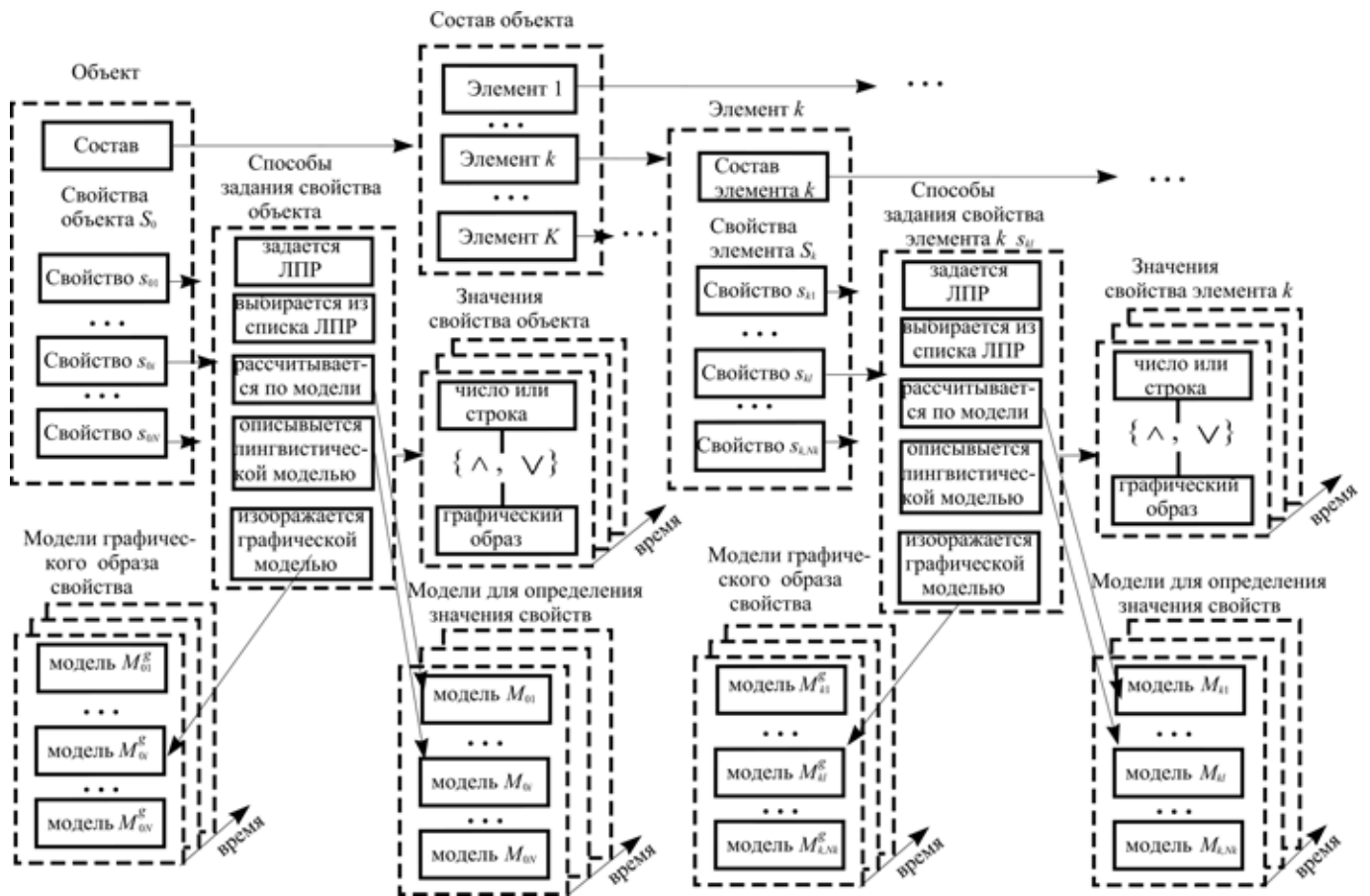


Рис. 1. Схема представления знаний об объекте культурно-исторического значения

При этом следует отметить, что способы задания свойства в течение времени могут меняться.

Элементами множества M_o являются модели для определения значений соответствующих свойств:

$$M_o = \{M_{o10}, \dots, M_{oit}, \dots, M_{oNT}\},$$

где M_{oit} — модель для определения значения i -го свойства объекта в t -й момент времени.

В связи с тем, что отдельное свойство может иметь числовое или строковое значение, то для его определения может быть использована либо математическая модель, определяющая значение свойства числом, либо лингвистическая модель, например модель концептуальной зависимости, определяющая значение свойства строкой символов.

Элементами множества M_o^g являются модели графических образов соответствующих свойств:

$$M_o^g = \{M_{o10}^g, \dots, M_{oit}^g, \dots, M_{oNT}^g\},$$

где M_{oit}^g — модель графического образа i -го свойства в t -й момент времени.

В свою очередь, каждый k -й элемент сложного информационного объекта O может быть описан аналогичным (1) способом:

$$O_k = \{St_k, S_k, Sp_k, M_k, M_k^g\}, \quad k = \overline{1, K},$$

где St_k — фрейм, описывающий структурный состав k -го элемента информационного объекта; S_k — фрейм, описывающий свойства, характерные для k -го элемента объекта; Sp_k — множество способов задания свойств k -го элемента объекта; M_k — множество моделей, позволяющих определить значения свойств, характерных для k -го элемента объекта; M_k^g — множество моделей графических образов свойств k -го элемента объекта, для которых он актуален (отдельные свойства могут не иметь графического образа);

$$s_{ki} = \{s'_{ki}, z'_{kit}, g'_{kit}\}, \quad i = \overline{1, N_k}; \quad t \in [0, T],$$

где s'_{ki} , z'_{kit} , g'_{kit} — соответственно наименование слота свойства s_{ki} k -го элемента объекта, его значение и графический образ; N_k — число свойств k -го элемента объекта; T — время жизненного цикла объекта.

Элементами множества Sp_k являются такие же термы, как и для Sp_o .

Для l -го свойства k -го элемента объекта, значение которого определяется в результате исполь-

зования аналитической или информационно-логической модели, предлагается модель M_{klT} :

$$M_k = \{M_{k10}, \dots, M_{klT}, \dots, M_{kN_kT}\}.$$

Для l -го свойства k -го элемента объекта, которое имеет графический образ, предлагается модель M_{klT}^g , реализованная на интервале времени $[0, T]$:

$$M_k^g = \{M_{k10}^g, \dots, M_{klT}^g, \dots, M_{kN_kT}^g\}.$$

Следует отметить, что информационный объект O и множество $O_k | k = \overline{1, K}$ (элементов объекта O) имеют аналогичную структуру. В связи с этим можно говорить о шаблоне для описания информационного объекта или его элементов.

Реализация модели

Реализацию модели рассмотрим на примере комплекса зданий (участка городской застройки), расположенного в городе Тамбове на углу улиц Советской и Коммунальной и являющегося объектом культурно-исторического значения.

В разные годы на протяжении последних 180 лет корпуса этого участка городской застройки были предназначены для проведения в них учебно-образовательных процессов. В настоящее время все учебные корпуса рассматриваемого объекта являются структурными подразделениями Тамбовского государственного технического университета (ТГТУ). Состав этого подразделения включает в себя следующие элементы (см. рис. 1): корпус технологического института (элемент 1); корпус кафедры автоматизированных систем и приборов (элемент 2); корпус кафедры информационных технологий в проектировании (элемент 3); кафе "Меридиан" (элемент 4); подсобные помещения (элемент 5). В свою очередь, каждый из перечисленных выше элементов подразделения ТГТУ распадается на отдельные подэлементы, которые в верхней части схемы (см. рис. 1) обозначены как "элемент k ". Например, в корпусе Технологического института размещается деканат заочного факультета, деканат факультета технической кибернетики и т. д. Это и будет являться фреймом, описывающим структурный состав объекта:

St_o , "настоящее время" = {"корпус технологического института", "корпус кафедры автоматизированных систем и приборов", "корпус кафедры информационных технологий в проектировании", "кафе Меридиан", "подсобные помещения"}.

Модели графических образов элементов представлены на рис. 2—5 (см. третью сторону обложки).

В качестве основного свойства рассматриваемого объекта культурно-исторического значения,

как отмечалось выше, принимается его основное функциональное назначение — учебно-образовательный процесс. Это свойство действительно соответствует всем элементам объекта и тем самым объединяет их. Значение этого свойства на протяжении рассматриваемого периода времени менялось. В частности, в период с 1825 по 1918 годы в зданиях рассматриваемого объекта проводилась гимназическая форма образования, с 1921 по 1958 годы учебный процесс соответствовал среднеспециальной форме образования, в третий временной период (с 1958 г. по наши дни) учебно-образовательный процесс ведется в форме высшего технического образования.

Каждое из перечисленных выше свойств имеет свои специфические способы задания, по-другому говоря, отличается, и это отличие, прежде всего, просматривается в учебных программах. Однако и последние не являются постоянно стабильными и имеют значительные расхождения, которые в те или иные годы диктовались политическими и экономическими особенностями. Принимая это во внимание, можно построить модель для определения значения свойств. В качестве примера рассмотрим значения свойства "направления деятельности" для разных периодов времени, представленные лингвистической моделью:

M_o , "направления деятельности", "1880-е годы" = {"организация и проведение учебного процесса", "жилищная функция", "проведение церковных служб", "библиотечная деятельность", "социальная служба питания", "медицинское обслуживание", "дополнительные спортивные занятия", "культурно-просветительная деятельность"};

M_o , "направления деятельности", "начало XX века" = {"организация и проведение учебного процесса", "жилищная функция", "проведение церковных служб", "библиотечная деятельность", "социальная служба питания", "медицинское обслуживание", "дополнительные спортивные занятия", "культурно-просветительная деятельность", "деятельность воскресной школы для жителей города", "патронажная деятельность учителей по вопросам работы других учебных заведений губернии", "организация религиозных служб для гимназистов и учителей не православного вероисповедания"}.

Значения свойств "общая численность учащихся" и "численность учащихся не православного вероисповедания" в различные периоды времени (от 1825 г. до 1913 г.) задаются регрессионными моделями, информация о которых приведена на рис. 6.

В ряде случаев при определении значений отдельных свойств, отражающих показатели исторического и культурного развития объекта, в особенности для исторического периода до 1900 года, имеется информация из различных источников,

Численность учащихся



Рис. 6. Функциональная зависимость изменения численности учащихся Тамбовской губернской мужской гимназии

которая может быть описана не числами, а словами естественного языка, которые в теории нечетких множеств называются термами. В этом случае для отображения лингвистической переменной необходимо определить четкие значения ее термов. Для решения этой задачи авторами использован подход, описанный в работе [2].

Таким образом, апробация предложенного подхода разработки модели информационного объекта для комплекса исторического и культур-

ного наследия осуществлена на примере комплекса зданий (участка городской застройки), расположенного в городе Тамбове на углу улиц Советской и Коммунальной.

Заключение

Предложенная авторами модель информационного объекта для комплекса исторического и культурного наследия, представленная в виде графовой структуры фреймов и включающая сведения о составе, свойствах системы и ее элементов, а также способах задания значений этих свойств, позволяет специалисту систематизировать всю информацию о реальном объекте, упорядочить ее хранение на электронных носителях и обеспечить эффективную обработку.

Список литературы

1. Минский М. Фреймы для представления знаний — [Электронный ресурс]. — URL: <http://a-future.ru/frejmy-dlya-predstavleniya-znaniy-m-minskij.html>.
2. Немтинов В. А., Горелов А. А. Методологические основы ретроспективного анализа объектов исторического и культурного наследия с использованием информационных технологий // КЛИО. 2008. № 3. С. 3—8.

УДК 004.043

М. Р. Богданов, канд. биол. наук, доц.,

А. В. Богданова, ст. преподаватель,

Ю. З. Габидуллин, аспирант,

Башкирский государственный педагогический университет им. М. Акмуллы, г. Уфа,

e-mail: junigobi@gmail.ru

Беспроводные сенсорные сети в экологии

Рассматриваются принципы построения беспроводных сенсорных сетей (БСС). Приводятся примеры использования БСС в экологических исследованиях, обсуждаются возникающие при этом трудности, а также пути их решения. Рассматриваются современные и перспективные сенсоры.

Ключевые слова: беспроводная сенсорная сеть, экология

Работа многих биологов и экологов, проводящих полевые исследования, сопряжена с выполнением большого числа рутинных операций. Пре-

жде всего это касается измерения различных физико-химических параметров атмосферы, воды и почвы. Проблема обычно усугубляется тем, что измерения желательно проводить на обширной территории или в течение длительного времени. Зачастую объекты измерений находятся в труднодоступных местах или сбор информации сопряжен с риском, причиной которого могут быть пожары, наводнения или землетрясения. При изучении диких животных нужно учитывать, что сам факт присутствия наблюдателя влияет на их поведение.

Для решения этих и других подобных задач можно использовать технологию беспроводных сенсорных сетей (БСС). Беспроводные сенсорные сети (WSN) состоят из вычислительных устройств — мотов, снабженных датчиками (сенсорами) и приемопередатчиками сигналов, работающими в заданном радиодиапазоне [1]. С помощью данной технологии можно эффективно решать задачи контроля и мониторинга. Работы по созданию БСС начались в середине 90-х годов прошлого века. За прошедшее время было создано несколько

интересных решений, которые с успехом применяются в области промышленной автоматизации, системах безопасности и т. д., например, технология ретранслируемой ближней радиосвязи IEEE 802.15.4/ZigBee. Однако в настоящее время еще не создана универсальная концепция беспроводных сенсорных сетей, вследствие чего реализация БСС во многом зависит от условий конкретного проекта.

В данном обзоре мы рассмотрим примеры использования БСС в экологических исследованиях, обсудим проблемы и перспективы развития данной технологии.

Почвенно-биологические исследования

Для почвоведов представляют интерес сведения о динамике температуры и влажности на разных глубинах, а также насколько эти показатели соотносятся с температурой воздуха и нормой осадков. Подобная информация не является срочной, поэтому нет необходимости получать ее в режиме он-лайн. Razvan Musaloiu-E с коллегами проводили экологические исследования в лесу близ Балтимора [2]. Ученые в феврале 2005 г. разместили в почве 10 узлов БСС, снабженных датчиками температуры и влажности. Расстояние между ноддами составляло 2 м. Измерения проводились раз в минуту и записывались в локальную флэш-память. Периодически исследователи обходили экспериментальные площадки и считывали накопленную информацию по протоколу IEEE 802.15.4 на ноутбук, играющий роль шлюза. Затем экспериментальные данные переносились в базу данных, где подвергались калибровке, сопоставлению с внешними источниками (температура и влажность воздуха, норма осадков) и визуализации. Результаты измерений доступны через сеть Интернет (рис. 1).

За 147 дней наблюдений было собрано 6 Мбайт данных. На рис. 2 и 3 приведены образцы измерений температуры и влажности. Изменения температуры хорошо согласуются с региональными тенденциями. Интересно сравнивать температуру на поверхности и на глубине 10 см. В то время как поверх-



Рис. 1. Архитектура системы сбора данных [2]

ность почвы иногда замерзает ниже нуля градусов, сама почва никогда не промерзает. Этого оказывается достаточно для того, чтобы почвенные беспозвоночные сохраняли активность. Осадки запускают циклы быстрого увлажнения и медленного высыхания. Узлы сети специально были раз-

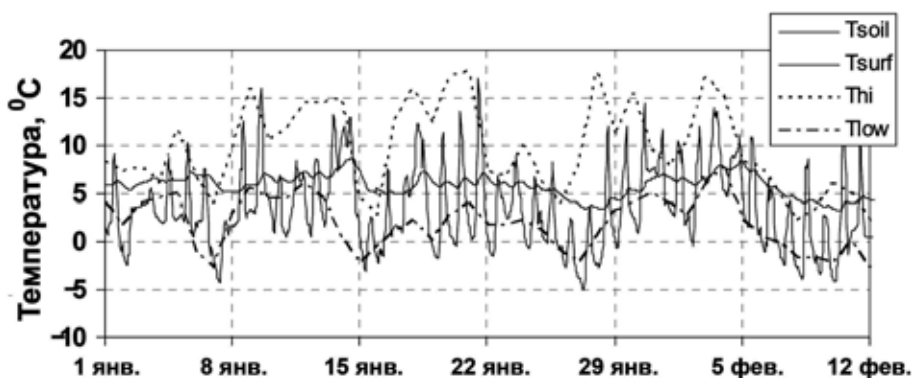


Рис. 2. Температура воздуха и почвы на протяжении шести недель. Каждая точка обозначает среднее значение за 6 ч; Tsoil — температура почвы на глубине 10 см; Tsurf — температура воздуха у земли; Thi и Tlow — максимальные и минимальные температуры для окрестностей Балтимора [2]

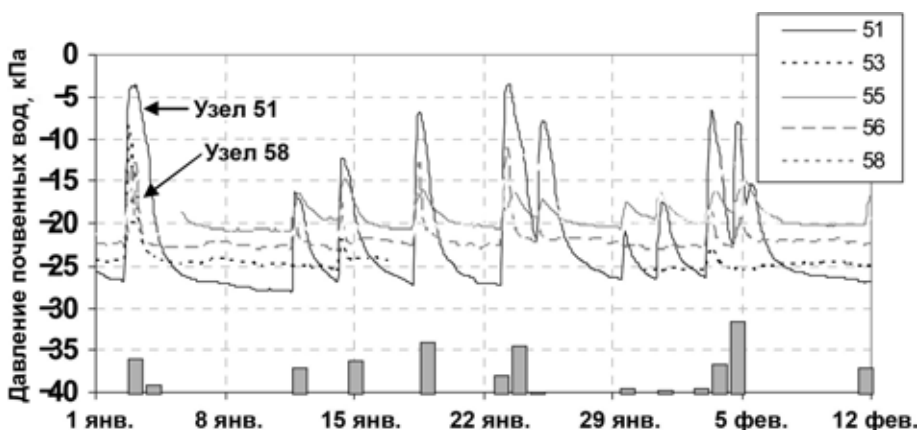


Рис. 3. Влажность почвы в течение 6 недель, измеренная пятью узлами. Каждая точка обозначает среднее значение за 6 ч. Колонки соответствуют дождям в Балтиморе, максимум пришелся на 5 февраля, когда выпало 25,4 мм осадков [2]

мещены на склоне и данные подтвердили наличие градиента влажности. Узел 51 (рис. 3), размещенный выше по склону, показал большие флуктуации, чем узел 51, расположенный ниже. Различия в показаниях индивидуальных сенсоров отражают пространственно-временную гетерогенность изучаемой области, что поможет лучше понять микробиологические процессы в почве, активность мезофауны и геохимические процессы, такие как почвенное дыхание.

Гидрологические исследования

Колебания природных условий в озерах протекают в гораздо более широких пределах, нежели в почвах. При этом изменения могут происходить очень быстро. В качестве примера можно привести высокогорное субтропическое озеро Юан Янг, расположенное на севере Тайваня. В данной местности нередки тайфуны. Оказывается, во время одного такого тайфуна в течение дня может выпасть 1 м осадков, а за два дня объем озера, имеющего глубину 4,5 м, может удвоиться [3]. Для ученых-лимнологов желательно осуществлять мониторинг озерных экосистем в режиме реального времени.

С помощью приборных буй (рис. 4) можно измерять скорость ветра, барометрическое давление, температуру воды на разных глубинах, а также содержание растворенного кислорода в воде. С помощью сети приборных буй можно отслеживать изменение биологических и физических параметров озер с разной временной шкалой. Изучая сезонную шкалу, можно ответить на вопрос, при каких условиях озера становятся источниками углерода в атмосфере? Изучение недельной динамики позволяет узнать, как озерный метаболизм отвечает на эпизодические явления, такие как ураганы и тайфуны. На минутной или часовой шкале видно, при каких условиях физические, а не биологические процессы влияют на динамику растворенного кислорода. Ведется работа по согласованию полевых измерений с трехмерной негидростатической моделью перемещения озерной воды в режиме реального времени для понимания механизмов циркуляции углерода на уровне всего озера при разных физических условиях в масштабе от нескольких минут до месяцев [3].

Многие озера располагаются в труднодоступных местах, поэтому для передачи данных в лабораторию можно использовать цифровой радиоканал с распределенным спектром. Ретрансляторы обеспечивают необходимую дальность (рис. 5).

При проведении экологических исследований на озере Юан Янг данные регистрировали один раз в течение 5–10 мин. В лабораторию инфор-



Рис. 4. Беспроводной буй на озере Юан Янг, Тайвань. Питается от солнечных батарей [3]



Рис. 5. Схема аппаратуры для беспроводной сенсорной сети. Для соединения последовательно опрашиваемых приборных буй с лабораторией используется цифровая радиосвязь с распределенным спектром. Скорость — 0,115 Мбит/с, частота — 900 МГц [3]

мацию передавали каждый час, где она хранилась в базе данных с Web-интерфейсом.

Беспроводные сенсорные сети позволяют собирать информацию во время штормов, пожаров и землетрясений. В качестве примера можно привести измерение динамики температуры на раз-

ных глубинах и интенсивность осадков на озере Юан Янг во время тайфуна (рис. 6, см. четвертую сторону обложки).

Мониторинг стихийных бедствий

В ряде случаев ученых интересуют события, происходящие только при определенных условиях, например, во время наводнения. В этом случае логично в обычное время проводить измерения за относительно большие интервалы времени, а при наступлении нужного события увеличить частоту измерений (рис. 7, см. четвертую сторону обложки).

Наблюдение за животными

Физическое присутствие наблюдателя влияет на поведение животных. И здесь также могут помочь сенсоры, установленные на значительном расстоянии от человека и позволяющие работать в режиме реального времени. С помощью звуковых сенсоров можно отслеживать перемещения наземных и морских млекопитающих, птиц, амфибий, летучих мышей и насекомых [4, 5]. По звуку можно оценить атмосферную турбулентность, интенсивность осадков, антропогенную нагрузку (шум самолетов, автомобилей и т. д.). Синхронная многоточечная акустическая сеть позволит воссоздать трехмерную модель распределения организмов. Локализовать местоположение отдельного животного можно с помощью акустической триангуляции [6]. Данные можно собирать, например, каждые полчаса и записывать в звуковой архив [6]. При анализе акустических сигналов возникает много вопросов. Например: звук принадлежит живому существу или он антропогенного происхождения (моторная лодка, выстрел ружья); какой именно птице (млекопитающему, амфибии или насекомому) принадлежит звук; какого типа звуковой сигнал (брачная песня амфибии или песня птиц, собирающихся на зимовку); какова связь между звуками и физическими (химическими, метеорологическими) характеристиками окружающей среды. Измерение звука в фиксированных местах в течение некоторого времени позволяет понять дневные и сезонные звуковые циклы экосистемы, а также сравнивать акустические сигнатуры разных местообитаний [6].

Работа с беспроводными сенсорными сетями имеет свою специфику. Хотелось бы обратить внимание на следующие аспекты их использования.

- ♦ Как правило, интересующие ученых события происходят чрезвычайно редко. Очень актуальной проблемой является автоматическое распознавание интересных событий и отбраковывание лишних данных, делать это нужно, по возможности, в режиме реального времени. Одним из решений проблемы является предва-

рительная классификация данных на месте (на узле БСС). Например, можно определить видовой состав животных в изучаемом районе и уже готовые данные переслать в лабораторию [3].

- ♦ При наступлении интересного события сенсор должен автоматически увеличить частоту измерений. Если исследователя интересует, например, ураган, пожар или землетрясение, то сенсор должен автоматически увеличивать частоту измерений при срабатывании нужного события [3].
- ♦ Анализируя данные с разной временной шкалой, можно получить много ценной информации. Проводя измерения с минутной шкалой, можно отследить отдельное животное, сезонная шкала поможет охарактеризовать экосистему. При этом желательно наряду с целевой информацией сенсора регистрировать также так называемые "метаданные", дополнительную информацию, такую как время измерения и географические координаты сенсора. Это в дальнейшем позволит анализировать данные ретроспективно и более глубоко [3].

Отдельно хотелось бы остановиться на сенсорах. В узлах БСС сенсоры преобразуют физические параметры (температуру, влажность, радиацию и т. д.), информацию о химических связях молекул или о биологических объектах в электрические сигналы, которые потом подвергаются аналогово-цифровому преобразованию, а иногда и статистической обработке. В настоящее время достигли технологической зрелости и получили широкое распространение физические датчики. Они дешевы, миниатюрны и долговечны. Химические сенсоры (азот и CO_2) пока еще очень дороги, критичны к питанию и не очень надежны. Фосфорные и белковые сенсоры пока находятся в стадии разработки [3]. При проведении экологических исследований, на наш взгляд, существуют две масштабные проблемы. Первая заключается в экспресс-методе определения широкого спектра анализируемых веществ. Современные химические сенсоры способны определять только одно соединение. Очень перспективной, на наш взгляд, является мультианалитическая беспроводная сенсорная платформа фирмы General Electric, способная определять целый спектр веществ в воде и воздухе [7]. Вторая проблема — определение вида животного. Решение этой задачи видится в использовании так называемых ДНК-чипов [8] (рис. 8, см. четвертую сторону обложки).

Список литературы

1. **Беспроводные** сенсорные сети [Электронный ресурс]: Википедия. URL: http://ru.wikipedia.org/wiki/Беспроводные_сенсорные_сети.

2. Musaloiu-E. R., Terzis A., Szlavecz K., Szalay A., Cogan J., Gray J. Life under your Feet: A Wireless Soil Ecology Sensor Network. [Электронный ресурс]: The Johns Hopkins University, Microsoft Research. URL: <http://lifeundervourfeet.org>.
3. Porter J., Arzberger P., Braun H.-W. et al. Wireless Sensor Networks for Ecology // BioScience. 2005. V. 55. № 7. P. 561–572.
4. Moore A., Miller J. R., Tabashnik B. E., Gage S. H. Automated identification of flying insects by analysis of wing beat frequencies // Journal of Economic Entomology. 1986. V. 79. № 6. P. 1703–1706.
5. Robbins C. S., Bystrak D., Geissler P. H. The Breeding Bird Survey: its first fifteen years, 1965–1979. US Department of Interior

Fish and Wildlife Service Resource Publication 157, Washington, DC. 1986.

6. Environmental Acoustic System [Электронный ресурс]: Remote Environmental Assessment Laboratory. URL: <http://environmental.cevl.msu.edu/acoustic/>.
7. Potyrailo R. A., Morris W. G. Multianalyte chemical identification and quantitation using a single radio frequency identification sensor // Analytical Chemistry. 2007. V. 79. P. 45–51.
8. NASA's Computing Information and Communications Technology (CICT) program [Электронный ресурс]: Official site of NASA. URL: www.cict.nasa.gov.

НЕЙРОИНФОРМАТИКА

УДК 004.8

В. Ю. Осипов, д-р техн. наук, вед. науч. сотр.,
Санкт-Петербургский институт
информатики РАН,
e-mail: osipov_vasily@mail.ru

Рекуррентная нейронная сеть с управляемыми синапсами

Предлагается рекуррентная нейронная сеть с управляемыми синапсами, обладающая расширенными возможностями по интеллектуальной обработке информации. Раскрыты возможности по реализации в ней управляемых пространственных сдвигов обрабатываемых сигналов. Уточнены правила запоминания информации на синапсах нейронов. Приведен анализ результатов вычислительного эксперимента.

Ключевые слова: рекуррентная нейронная сеть, управляемые синапсы, структура, правила, интеллектуальная обработка информации

Введение

Поиск путей расширения функциональных возможностей нейронных сетей по интеллектуальной обработке информации в настоящее время представляет большой научный и практический интерес. Наличие их позволит создать высокоинтеллектуальные системы и машины, существенно облегчающие познавательную и созидательную деятельность человека в различных областях.

Один из перспективных путей такого совершенствования — развитие рекуррентных нейронных сетей.

Известны рекуррентные нейронные сети [1–7], применяемые для ассоциативного запоминания информации, нелинейного прогнозирования и моделирования, обработки речевых и других сигналов, распознавания процессов, управления различными объектами. Среди них ассоциативные

сети (Хопфилда, Хемминга, ВАН), рекуррентные сети на базе персептрона (RMLP, RTRN, Эльмана) и другие.

К недостаткам известных рекуррентных сетей относятся существенно ограниченные возможности по обработке различных видов сигналов, их распознаванию, запоминанию, извлечению из памяти сети. Имеет место быстрое размывание структур сигналов в сетях. Эти сети не позволяют в полной мере реализовать искусственное мышление и обеспечить решение одной и той же сетью различных интеллектуальных задач.

Это обусловлено как несовершенством архитектур рекуррентных нейронных сетей, так и моделей их элементов.

Предлагается новая рекуррентная нейронная сеть с управляемыми синапсами и расширенными возможностями по интеллектуальной обработке информации.

Постановка задачи

Полагается, что в искомую рекуррентную сеть должны подаваться структурно-сложные (акустические, электрические, магнитные, тепловые, видео, механические и другие) сигналы, предварительно разложенные на составляющие в базисе, согласованном с ее входным слоем. При этом каждая составляющая перед подачей в сеть должна быть преобразована в последовательность единичных образов с частотой повторения, как предварительно заданной функции от ее амплитуды. В итоге на вход искомой сети должны подаваться последовательные совокупности единичных образов, несущие всю информацию о входных сигналах. Информация о частотных и пространственных характеристиках составляющих сигнала в рассматриваемом случае закрепляется за номерами формируемых последовательностей единичных образов. Фазовые характеристики составляю-

щих однозначно связаны с задержками единичных образов.

При обработке входных воздействий на выходе искомой нейронной сети должны иметь место последовательные совокупности единичных образов, несущие информацию о сигналах, вызванных из памяти, связанных с входными сигналами. В частном случае это может быть информация, смещенная во времени, о входных сигналах, очищенных от ошибок и помех.

Предусматривается, что по последовательностям единичных образов на выходе нейронной сети путем обратного преобразования могут быть воспроизведены соответствующие им исходные сигналы.

Искомая рекуррентная нейронная сеть должна запоминать входные сигналы в своей памяти, ассоциировать их с ранее запомненными воздействиями и функционировать в соответствии с вскрытыми закономерностями внешнего мира. Она должна позволять решать различные интеллектуальные задачи без изменения своей физической структуры.

Необходимо разработать рекуррентную нейронную сеть, удовлетворяющую этим требованиям.

Структура нейронной сети

В интересах разработки такой сети учтем свойства, характерные для биологических нейронных сетей. Среди них обратим внимание на наличие у биологических нейронов состояний невосприимчивости к внешним воздействиям после возбуждения. Время нахождения биологических нейронов в этих состояниях существенно больше времени задержки в них сигналов. Кроме этого в биологической нейронной сети между электрическими зарядами существуют силы притяжения и отталкивания, которые влияют на направление потоков импульсов.

Принимая это во внимание, в качестве нейронной сети для интеллектуальной обработки сигналов предлагается использовать сеть с обратными связями, замыкающими многослойные контуры с временем задержки единичных образов меньше времени невосприимчивости нейронов после их возбуждения. При передаче совокупностей единичных образов в нейронной сети от слоя к слою рекомендуется сдвигать их вдоль слоев с учетом текущих состояний последних. Это достижимо за счет специального управления синапсами нейронов.

Минимальная конфигурация рекуррентной нейронной сети, удовлетворяющей всем этим требованиям, может иметь вид, приведенный на рис. 1. Эта сеть представляет собой своеобразное интеллектуальное нейросетевое ядро. Особенность этого ядра в структурном плане, а также других рекур-

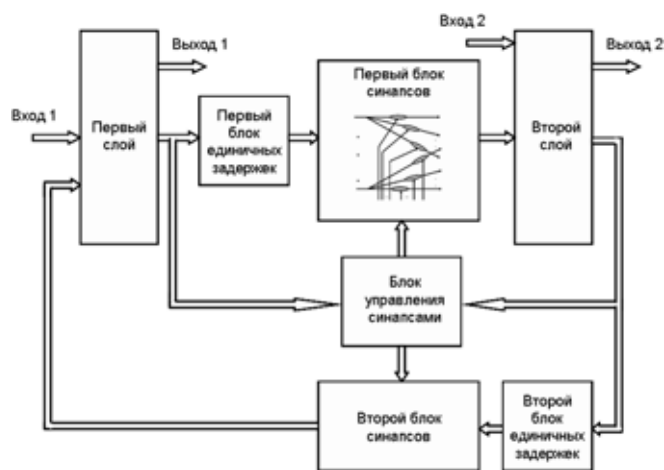


Рис. 1. Структура нейронной сети

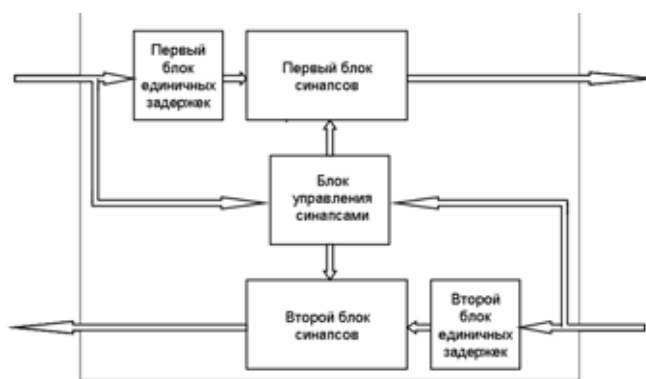


Рис. 2. Модуль связи слоев нейронной сети

рентных нейронных сетей, построенных на его основе, в том, что связь двух смежных слоев осуществляется с использованием специального модуля (рис. 2).

В этом модуле блоки единичных задержек обеспечивают резерв времени на выработку и реализацию управлений синапсами. Как будет показано ниже, модуль позволяет осуществлять прямую и обратную передачу сигналов между слоями с сохранением их структуры, совершенствовать их запоминание и ассоциирование.

Поясним правила обработки информации в предложенной сети с учетом управления синапсами нейронов.

Правила обработки информации

При поступлении на первый слой нейронной сети очередной совокупности единичных образов с учетом обратных результатов преобразования на ее выходе формируется новая совокупность, несущая информацию как о входных сигналах, так и о ранее запомненных, циркулирующих в сети. Затем каждый единичный образ из сформированной совокупности задерживается в первом блоке

единичных задержек и поступает на свою совокупность синапсов, связывающих каждый возбужденный нейрон первого слоя в общем случае со всеми нейронами второго слоя. На синапсах осуществляется запоминание поступивших единичных образов и считывание информации о предыдущих воздействиях. В зависимости от управляющих воздействий со стороны блока управления синапсами на них реализуется пространственный сдвиг поступающих на второй слой совокупностей единичных образов. На втором слое осуществляются ассоциации между элементами входного сигнала, а также между ним и запомненными сетью сигналами. На выходе этого слоя формируются новые совокупности единичных образов, несущие информацию о запомненных сигналах, связанных с входным сигналом. Эти совокупности после задержки аналогичным образом обрабатываются во втором блоке синапсов и поступают на первый слой, на котором осуществляются ассоциации между ними и другими запомненными сигналами. Затем процесс повторяется. В результате на первом и втором выходах сети формируются последовательные совокупности единичных образов, несущие информацию о сигналах, вызванных из памяти, связанных с входными воздействиями.

Для запоминания информации на синапсах и осуществления пространственных сдвигов совокупностей единичных образов используются следующие правила. Вес каждого синапса необходимо определять через произведение его весового коэффициента $k_{ij}(t)$ и функции ослабления $\beta(r_{ij}(t))$. Весовой коэффициент каждого синапса изменяется при поступлении на него единичного образа. Это изменение коэффициента закрепляется только при возбуждении через синапс, в общем случае совместно с другими воздействиями, другого j -го нейрона. В противном случае изменение весового коэффициента синапса не закрепляется. В формализованном виде это правило может быть записано как

$$k_{ij}(t) = \begin{cases} \frac{2}{1 + \exp(-\gamma g_{ij}(t))} - 1, & \text{если } j\text{-й нейрон} \\ & \text{возбуждается,} \\ k_{ij}(t) = k_{ij}(t - \Delta t), & \text{в противном случае,} \end{cases}$$

где $g_{ij}(t)$ — число запомненных синапсом единичных образов на момент времени t ; γ — положительный коэффициент; $k_{ij}(t - \Delta t)$ — значение весового коэффициента до поступления на синапс очередного единичного образа.

В качестве функции ослабления предлагается использовать функцию, значение которой зависит от геометрического расстояния $r_{ij}(t) =$

$= \sqrt{x_{ij}^2(t) + y_{ij}^2(t)}$ между нейронами взаимодействующих слоев. Для нейронов, имеющих одни и те же порядковые номера во взаимодействующих слоях, эта функция равна единице, в других случаях — существенно меньше ее. Управление пространственными сдвигами совокупностей единичных образов при передаче их от слоя к слою в этом случае осуществимо путем сдвигов номеров нейронов одного слоя относительно другого. Необходимость и целесообразность таких сдвигов обусловлена следующими аспектами. Совместно с предложенными правилами запоминания информации на синапсах это позволяет во-первых, устранить размывание структур сигналов в нейронной сети вследствие исключения наложения обратных результатов распознавания на прямые воздействия, во-вторых, обеспечить более высокий уровень параллельной обработки информации в нейронной сети. Рассматриваемые сдвиги позволяют также приводить сигналы, поступившие в сеть, к одним моментам времени, обеспечить их ассоциацию между собой и запомненными сигналами. В-третьих, предоставляется возможность переключения нейронной сети с одной "мысли" на другую. Под "мыслью" в данном случае понимается поток совокупностей единичных образов, несущий информацию о конкретном явлении, процессе, предмете и т. п.

В общем виде задача управления синапсами может быть сформулирована следующим образом. Необходимо найти значение пространственного сдвига $\Delta x(t)$ совокупностей единичных образов по координате x как функцию $\Delta x(t) = \varphi(S_1(t), S_2(t))$ от состояний $S_1(t), S_2(t)$ первого и второго слоев, при которой обеспечивается минимум конфликтов $G_{0_{1(2)}}(t)$ между их нейронами,

$$G_{0_1}(t) = \sum_{i=j \in \Omega} s_{1i}(t) s_{2j}^*(t, \Delta x(t)) \rightarrow \min,$$

или

$$G_{0_2}(t) = \sum_{i=j \in \Omega} s_{2j}(t) s_{1i}^*(t, \Delta x(t)) \rightarrow \min,$$

где функция $s_{1i(2j)}(t) = 1$, если i -й (j -й) нейрон первого (второго) слоя находится в состоянии возбуждения, в противном случае — $s_{1i(2j)}(t) = 0$; $s_{1i(2j)}^*(t, \Delta x(t)) = 1$, если i -й (j -й) нейрон первого (второго) слоя находится в состоянии невосприимчивости, в других случаях — $s_{1i(2j)}^*(t, \Delta x(t)) = 0$.

В частном случае для принятия решений о сдвигах в блоке управления синапсами анализируются состояния первого и второго слоев сети. В случае отсутствия препятствий для передачи совокупностей единичных образов от одного слоя к другому по коротким связям (между нейронами с одними порядковыми номерами), сдвиги не осу-

ществляются. Когда же соответствующие нейроны воспринимающего слоя находятся в состояниях невосприимчивости к воздействиям, может реализовываться однонаправленный пространственный сдвиг этих совокупностей, например, на постоянную величину вправо. При необходимости сдвига совокупностей за пределы слоя сдвиг осуществляется от его начала. В качестве количественного критерия принятия решения о сдвиге может выступать превышение некоторого порога отношением числа соответствующих нейронов воспринимающего слоя, находящихся в состояниях невосприимчивости, к числу возбужденных нейронов передающего слоя.

Все единичные образы, поступившие на один и тот же нейрон по различным синапсам, суммируются. При превышении этой суммы заданного порога возбуждения нейрона он возбуждается и на его выходе формируется единичный образ. Затем сумма обнуляется, а сам нейрон переходит в состояние невосприимчивости входных сигналов. В нем он находится заданное время, одинаковое для всех нейронов сети, которое в нашем случае больше суммарной задержки единичных образов во всех блоках, входящих в двухслойный многопучковый контур нейронной сети.

Результаты вычислительного эксперимента

Осуществлялось моделирование предложенной нейронной сети в среде MatLab. Результаты моделирования подтвердили наличие у нее свойств, удовлетворяющих предъявленным требованиям.

Входные сигналы в нейронной сети не разрушались. При этом они запоминались не только в долговременной памяти на синапсах, но и в оперативной памяти через состояния нейронов. Они позволяли извлекать из памяти сети связанные с ними сигналы. По последовательностям совокупностей единичных образов на любом из выходов нейронной сети можно было воспроизвести исходные воздействия. Простой пример смены состояний предложенной сети при обработке трех входных совокуп-

ностей единичных образов с информационной структурой "К__О__Т", вводимых в сеть через 1—3 поля первого слоя, приведен в таблице.

В таблице приняты обозначения: 0 — нейроны в состояниях готовности к приему воздействий; 1 — нейроны возбуждены; x — нейроны в состояниях невосприимчивости.

Сдвиги совокупностей единичных образов в этом примере осуществляются вправо на четыре позиции. Заметим, что сдвиг имеет место там, где слева от сдвинутой совокупности имеет место совокупность нейронов, находящихся в состояниях x — невосприимчивость, продолжительностью в два такта. Сигнал с такой сети может быть снят, если наблюдать только за одной из групп полей первого или второго слоя. Например, если анализировать поля 5—7 или 9—11 на протяжении 12 тактов, то можно прочитать слово "КОТ", которое было введено в сеть. При продольном анализе со-

№ такта	Состояние первого слоя	№ такта	Состояние первого слоя
1		7	
2		8	
3		9	
4		10	
5		11	
6		12	

стояний слоев нейронной сети видно, что сигналы, поступившие в нее в различные моменты времени, приводятся также к одному моменту и могут ассоциировать друг друга.

Аналогичным образом с использованием предложенной нейронной сети можно обрабатывать различные структурно-сложные сигналы.

Выводы

1. Предложенная нейронная сеть с управляемыми синапсами представляет собой интеллектуальное нейросетевое ядро, обладающее расширенными возможностями по обработке разнородной информации.

2. Управление синапсами позволяет устранить разрушение в сети структур обрабатываемых сигналов, совершенствовать их запоминание и ассоциирование.

3. На базе этой сети могут быть разработаны перспективные ассоциативные машины, реали-

зующие интеллектуальную обработку информации в широком смысле.

Список литературы

1. **Галушкин А. И.** Теория нейронных сетей. Кн. 1: учеб. пособие для вузов / Общая ред. А. И. Галушкина. М.: ИПРЖР, 2000. 416 с.
2. **Осовский С.** Нейронные сети для обработки информации / Пер. с англ. И. Д. Рудницкого. М.: Финансы и статистика, 2002. 344 с.
3. **Jim-Shih Liaw, Berger T. W.** Dynamic synapse for signal processing in neural networks. United States Patent. No.: US 6,643,627 B2. Nov. 4, 2003.
4. **Хайкин С.** Нейронные сети: полный курс. 2-е изд.: Пер. с англ. М.: Вильямс, 2006. 1103 с.
5. **Макаров И. М., Лохин В. М., Манько С. В., Романов М. П.** Искусственный интеллект и интеллектуальные системы управления. М.: Наука, 2006. 333 с.
6. **Карпов В. Э., Вальцев В. Б.** Динамическое планирование поведения робота на основе сети "интеллектуальных" нейронов // Искусственный интеллект и принятие решений. 2009. № 2.
7. **Алакоз Г. М., Светлов Р. В.** Машина Тьюринга как ассоциативная нейроЭВМ // Нейрокомпьютеры: разработка, применение. 2009. № 5.

УДК 519.671

О. В. Крючин, аспирант,
ТГУ им. Г. Р. Державина, г. Тамбов,
e-mail: kryuckov@gmail.com

Нейросетевой симулятор с автоматическим подбором активационных функций

Описаны алгоритм обучения искусственной нейронной сети с автоматическим подбором активационных функций нейронов, а также программа, реализующая этот алгоритм и способы его ускорения.

Ключевые слова: нейронные сети, параллельные алгоритмы

Введение

Известно, что возможности искусственных нейронных сетей (ИНС) при моделировании объектов различной природы зависят от их структуры, под которой обычно понимают число скрытых слоев, число нейронов в этих слоях, имеющуюся систему связей между нейронами и их активационные функции. Правильный выбор указанных параметров в значительной степени позволяет достигать адекватности ИНС-модели реальному объекту.

В настоящее время не существует общего алгоритма определения структуры ИНС, подходящего

для каждой рассматриваемой проблемы. Часто такую структуру выбирают методом "проб и ошибок", который отнимает у исследователя много времени. В некоторых программных ИНС-оболочках предусмотрена визуализация построения структуры сети.

Целью данной работы является разработка симулятора для компьютерного моделирования на основе ИНС с автоматическим подбором активационных функций [1].

Перебор вариантов активационной функции нейрона

Универсальный математический нейрон состоит из двух частей — сумматора и активационной функции (рис. 1). Сумматор получает на вход произведения выходных сигналов других нейронов (на рисунке x_0, x_1, x_2) и значений весовых коэффициентов (на рисунке w_0, w_1, w_2), а также величину импульса. В данном случае имеются два импульса (дополнительные весовые коэффициенты), названные внутренним (p_I) и внешним (p_E), которые различаются тем, что только внешний импульс может подбираться наряду с весовыми коэффициентами. После того, как сумматор вычисляет значение, оно умножается на коэффициент и становится аргументом активационной функции. Значение активационной функции является выходным значением нейрона (y). Таким

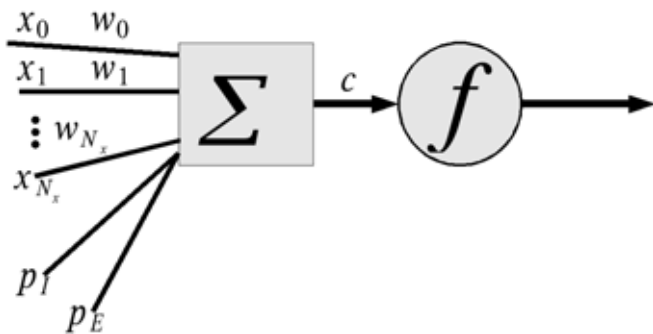


Рис. 1. Модель математического нейрона

образом, выходное значение i -го нейрона можно выразить следующей формулой:

$$y_i = f\left(c_f\left(\sum_j x_j w_{j,i} + p_I + p_E\right)\right), \quad (1)$$

где x_j — выходное значение j -го нейрона; $w_{j,i}$ — значение весового коэффициента, соединяющего j -й и i -й нейроны; c_f — коэффициент активационной функции; p_I — внутренний импульс; p_E — внешний импульс; f — активационная функция.

Активационные функции могут быть следующих видов:

- линейная ($y = x$);
- квадратичная ($y = x^2$);
- кубическая ($y = x^3$);
- функция четвертой степени ($y = x^4$);
- синусоида ($y = \sin(x)$);
- косинусоида ($y = \cos(x)$);
- тангенсная ($y = \operatorname{tg}(x)$);
- котангенсная ($y = \operatorname{ctg}(x)$);
- четыре вида сигмоиды ($y = \frac{1}{1+x}$, $y = \frac{1}{1+e^x}$,

$$y = \frac{1}{1+e^{-x}}, y = \frac{1}{1+e^{|x|});$$

- пороговая функция.

Все эти функции, за исключением пороговой, являются простыми. Пороговая же может быть различной. Например, предложенная МакКалок и Питсом функция

$$y = \begin{cases} 0, & x \leq 0; \\ 1, & x > 0 \end{cases}$$

используется в классическом нейроне. Развита полная теория, позволяющая синтезировать произвольные логические схемы на основе функции нейронов с такой нелинейностью [2]. В сети Хемминга используется немного другой вид пороговой функции:

$$y = \begin{cases} 0, & x \leq 0; \\ x, & x > 0. \end{cases}$$

Существуют и иные модификации.

Таким образом, универсальный вид пороговой функции состоит из пяти элементов (значения

порога, верхнего и нижнего стандартного значений, а также верхнего и нижнего флага, показывающих, следует ли заменять входное значение на стандартное):

$$y = \begin{cases} \hat{y}_1 & \text{при } b_{C1} = 1 \text{ и } x < h_t; \\ x & \text{при } b_{C1} = 1 \text{ и } x \geq h_t; \\ \hat{y}_0 & \text{при } b_{C0} = 1 \text{ и } x \geq h_t; \\ x & \text{при } b_{C0} = 0 \text{ и } x < h_t, \end{cases} \quad (2)$$

где \hat{y}_1 , \hat{y}_0 — верхнее и нижнее стандартные значения; b_{C1} , b_{C0} — верхний и нижний флаги замены; h_t — значение порога.

То есть значение функции принимает входное значение в двух случаях:

- если входное значение меньше порога ($x < h_t$) и нижний флаг замещения не равен нулю ($b_{C0} \neq 0$);
- если входное значение не меньше порога ($x \geq h_t$) и верхний флаг замещения равен нулю ($b_{C1} = 0$).

В противном случае выходное значение принимает либо верхнее, либо нижнее стандартные значения (в зависимости от того, больше или меньше входное значение, чем пороговое). В случае функции МакКалок—Питса $h_t = 0$, $b_{C1} = 1$, $b_{C0} = 1$, $\hat{y}_1 = 1$, $\hat{y}_0 = 0$, а в случае функции Хемминга $h_t = 0$, $b_{C1} = 0$, $b_{C0} = 1$, $\hat{y}_0 = 0$.

Исходя из вышесказанного можно заключить, что существует два вида нейрона: с простой и пороговой активационной функциями. Первый из них имеет четыре подбираемых элемента, второй — шесть. Из них, за исключением номера простой активационной функции, значения которого является перебираемым, значения всех являются вещественными числами.

В переборе вариантов могут участвовать не все элементы, а лишь некоторые, поэтому выразить число состояний нейрона затруднительно. Например, в случае, когда в переборе участвуют коэффициент и оба импульса, значения которых перебираются в пределах $[-1; 1]$ с шагом 0,5, нейрон имеет 625 состояний.

На рис. 2 представлена блок-схема алгоритма изменения состояния нейрона. Некоторые из условий обозначены числами:

1. Проверка, участвует ли в переборе коэффициент активационной функции c_f и в случае участия — меньше ли он верхнего предела $s_{cf,u}$ (выполняется ли условие $c_f < s_{cf,u}$). При выполнении условия результат положителен и коэффициент увеличивается на величину шага $s_{cf,s}$ ($c_f := c_f + s_{cf,s}$), во всех других случаях — отрицателен.

2. Проверка, участвует ли в переборе внутренний импульс и в случае участия — выполняется ли условие $p_I < s_{pI,u}$, т. е. не достигло ли значение

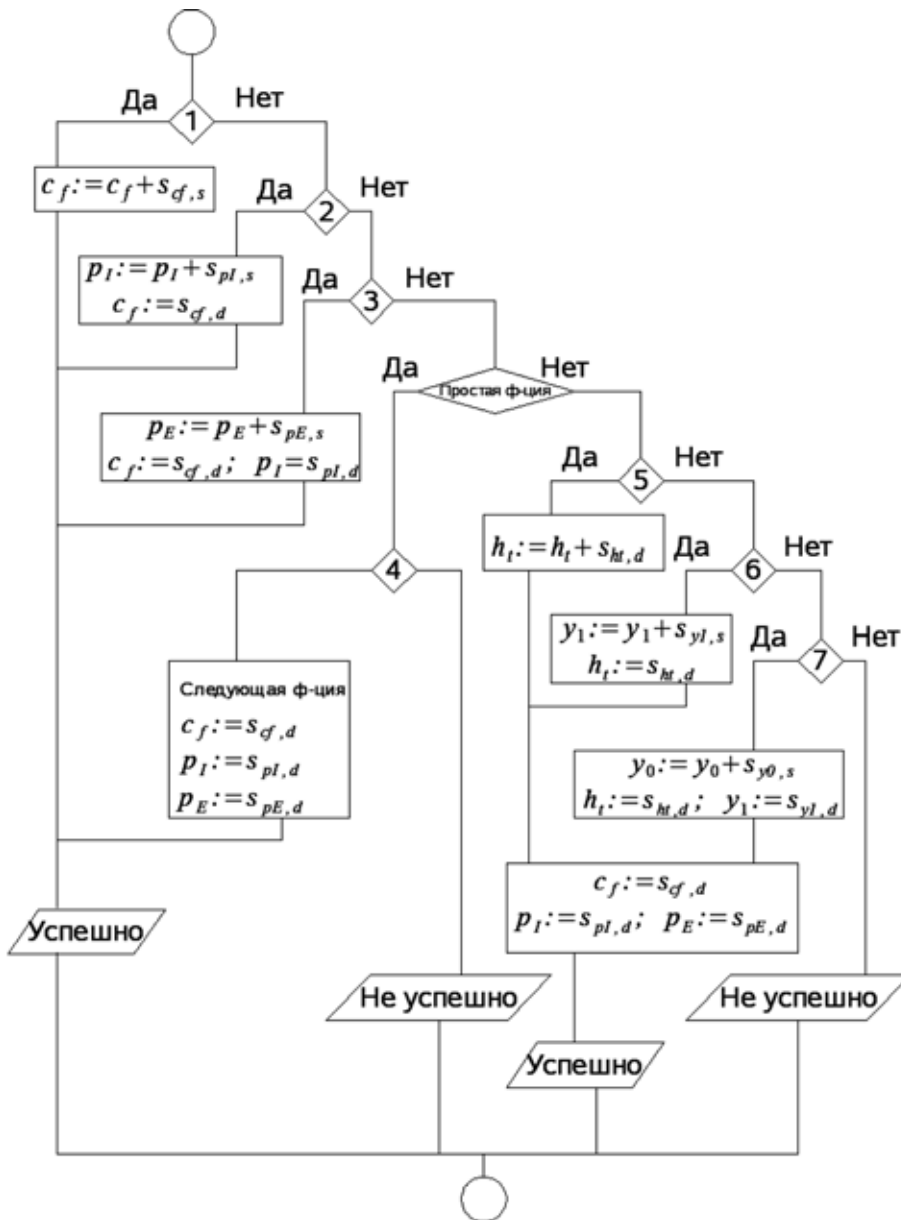


Рис. 2. Блок-схема изменения одного из элементов нейрона

внутреннего импульса p_I верхнего предела $s_{pI, u}$. В случае выполнения условия результат положителен и внутренний импульс увеличивается ($p_I := p_I + s_{pI, s}$), а коэффициент активационной функции c_f устанавливается начальным значением $s_{cf, d}$ ($c_f := s_{cf, d}$), во всех других случаях результат отрицателен.

3. Проверка, участвует ли в переборе внешний импульс и в случае участия — выполняется ли условие $p_E < s_{pE, u}$, т. е. не достигло ли значение внешнего импульса p_E верхнего предела $s_{pE, u}$. В случае выполнения условия результат положителен, следовательно, внешний импульс увеличивается ($p_E := p_E + s_{pE, s}$), а внутренний импульс p_I и коэффициент активационной функции c_f уста-

навливаются начальными значениями ($p_I := s_{pI, d}$, $c_f := s_{cf, d}$), во всех других случаях результат отрицателен.

4. Проверка, участвует ли в переборе простая активационная функция и в случае участия — является ли текущая функция не последней из перебираемых. В случае выполнения этого условия результат положителен (устанавливается следующая активационная функция, а коэффициент активационной функции и импульсы принимают начальные значения ($c_f := s_{cf, d}$, $p_I := s_{pI, d}$, $p_E := s_{pE, d}$)), во всех других случаях — отрицателен.

5. Проверка, участвует ли в переборе пороговое значение и в случае участия — выполняется ли условие $h_t < s_{ht, u}$ (не достигло ли пороговое значение верхнего предела). В случае выполнения условия результат положителен (пороговое значение увеличивается ($h_t := h_t + s_{ht, s}$), а коэффициент активационной функции и импульсы принимают начальные значения ($c_f := s_{cf, d}$, $p_I := s_{pI, d}$, $p_E := s_{pE, d}$)), во всех других случаях — отрицателен.

6. Проверка, участвует ли в переборе верхнее стандартное значение и в случае участия — выполняется ли условие $y_1 < s_{y1, u}$ (не достигло ли верхнее стандартное значение y_1 своего верхнего предела $s_{y1, u}$). В случае выполнения условия результат положителен (коэффициент активационной функции, пороговое

значение и импульсы принимают начальные значения ($c_f := s_{cf, d}$, $h_t := s_{ht, d}$, $p_I := s_{pI, d}$, $p_E := s_{pE, d}$), а верхнее стандартное значение y_1 увеличивается на величину шага $s_{y1, s}$), во всех других случаях — отрицателен.

7. Проверка, участвует ли в переборе нижнее стандартное значение и в случае участия — выполняется ли условие $y_0 < s_{y0, u}$ (не достигло ли нижнее стандартное значение y_0 своего верхнего предела $s_{y0, u}$). В случае выполнения условия результат положителен, во всех других случаях — отрицателен. В случае положительного результата нижнее стандартное значение увеличивается ($y_0 := y_0 + s_{y0, s}$), а верхнее, коэффициент активационной функции, пороговое значение и импульсы уста-

навливаются начальным значением ($y_1 := s_{y1}, d, c_f := s_{cf}, d, h_t := s_{ht}, d, p_I := s_{pI}, d, p_E := s_{pE}, d$).

Во всех условиях положительный результат означает линию влево, отрицательный — вправо. Успешное завершение алгоритма означает, что состояние нейрона изменено, неуспешное — что текущее состояние последние из возможных, и, следовательно, изменить его нельзя.

Автоматический подбор активационных функций

Задача обучения ИНС состоит в минимизации функции

$$\varepsilon = \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \|y_{i,j} - d_{i,j}\| = \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \|F(x_i, w, \mu)_j - d_{i,j}\| \quad (4)$$

где $d_{i,j}$ — выходные значения моделируемого объекта; $y_{i,j}$ — выходные значения ИНС; N — число строк в обучающей выборке; P — число выходов сети; x_i — выходные значения; w — вектор коэффициентов синаптических связей; μ — вектор состояний нейронов (видов активационных функций).

Для успешного обучения необходимо подобрать значения μ и w таким образом, чтобы минимизировать невязку. Сделать это можно следующим образом: перебрать все комбинации состояний нейронов $\mu = (\mu(f_0), \mu(f_1), \dots, \mu(f_{I_{N-1}}))$, подбирая для каждой комбинации весовые

коэффициенты. То есть вначале следует установить начальное состояние всех нейронов $\mu = (\mu_0, \mu_0, \dots, \mu_0)$ и, используя какой-либо алгоритм подбора весовых коэффициентов, подобрать значения синаптических связей и вычислить значение невязки, затем по описанному выше алгоритму изменить состояние первого нейрона и повторить эту операцию. Таким образом, состояния нейронов изменяются следующим образом:

$$\begin{aligned} \mu &= (\mu_0, \mu_0, \dots, \mu_0, \mu_0); \\ \mu &= (\mu_1, \mu_0, \dots, \mu_0, \mu_0); \\ \mu &= (\mu_2, \mu_0, \dots, \mu_0, \mu_0); \\ &\dots \\ \mu &= (\mu_{N(\mu)-1}, \mu_0, \dots, \mu_0, \mu_0); \\ \mu &= (\mu_0, \mu_1, \dots, \mu_0, \mu_0); \\ \mu &= (\mu_1, \mu_1, \dots, \mu_0, \mu_0); \\ &\dots \\ \mu &= (\mu_0, \mu_0, \dots, \mu_0, \mu_{N(\mu)-1}); \\ \mu &= (\mu_1, \mu_0, \dots, \mu_0, \mu_{N(\mu)-1}); \\ \mu &= (\mu_2, \mu_0, \dots, \mu_0, \mu_{N(\mu)-1}); \\ &\dots \\ \mu &= (\mu_0, \mu_{N(\mu)-1}, \dots, \mu_{N(\mu)-1}, \mu_{N(\mu)-1}); \\ &\dots \\ \mu &= (\mu_{N(\mu)-1}, \mu_{N(\mu)-1}, \dots, \mu_{N(\mu)-1}, \mu_{N(\mu)-1}). \end{aligned}$$

Для каждого набора состояний подбираются значения весовых коэффициентов и вычисляется значение целевой функции (4), и, таким образом, после перебора всех вариантов получается набор значений погрешности. Минимальная из них соответствует наилучшему состоянию сети.

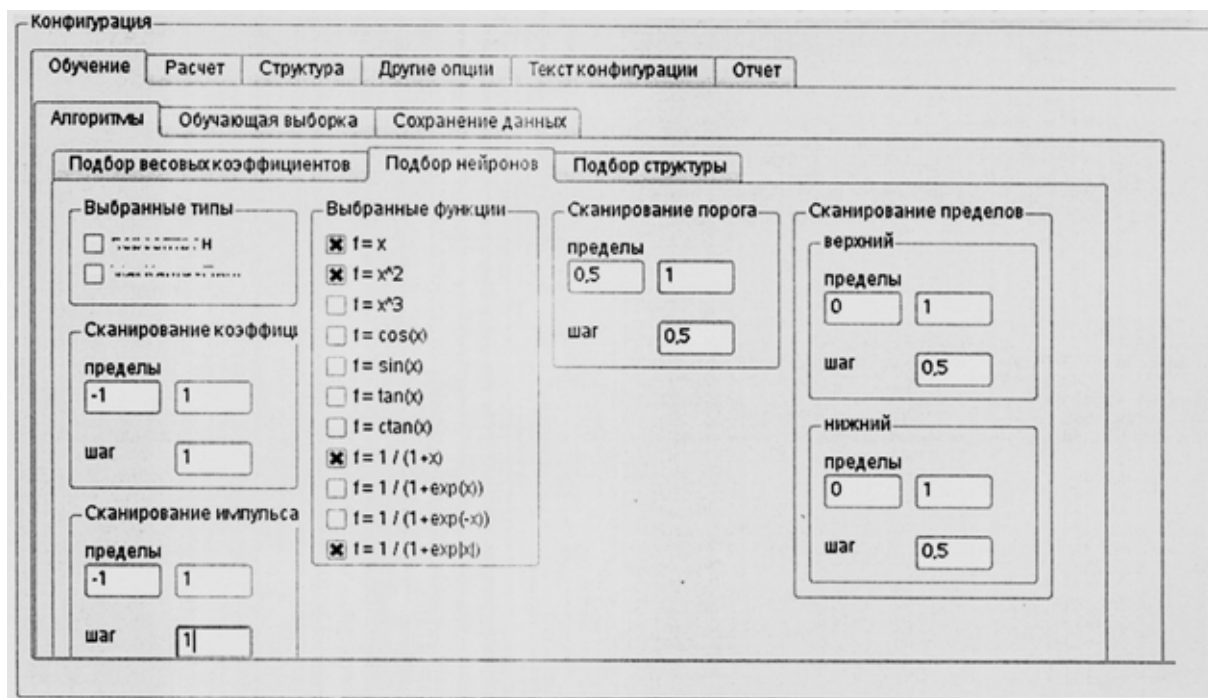


Рис. 3. Интерфейс конфигурации перебора нейронов

Разработка программного обеспечения

Приведенный выше алгоритм реализован в виде программной оболочки, интерфейс которой показан на рис. 3–5. На рис. 3 представлены конфигурация перебора нейронов, где можно выбрать перебираемые виды активационных функций, и используемые элементы — коэффициент активационной функции, импульс, значе-

ние порога и верхний и нижний стандартные значения. На рис. 4 показана конфигурация алгоритма подбора весовых коэффициентов, используемого для подбора значений синаптических связей. Программа позволяет визуально отображать состояния нейронов (рис. 5). В зависимости от состояния нейроны отображаются различным цветом.

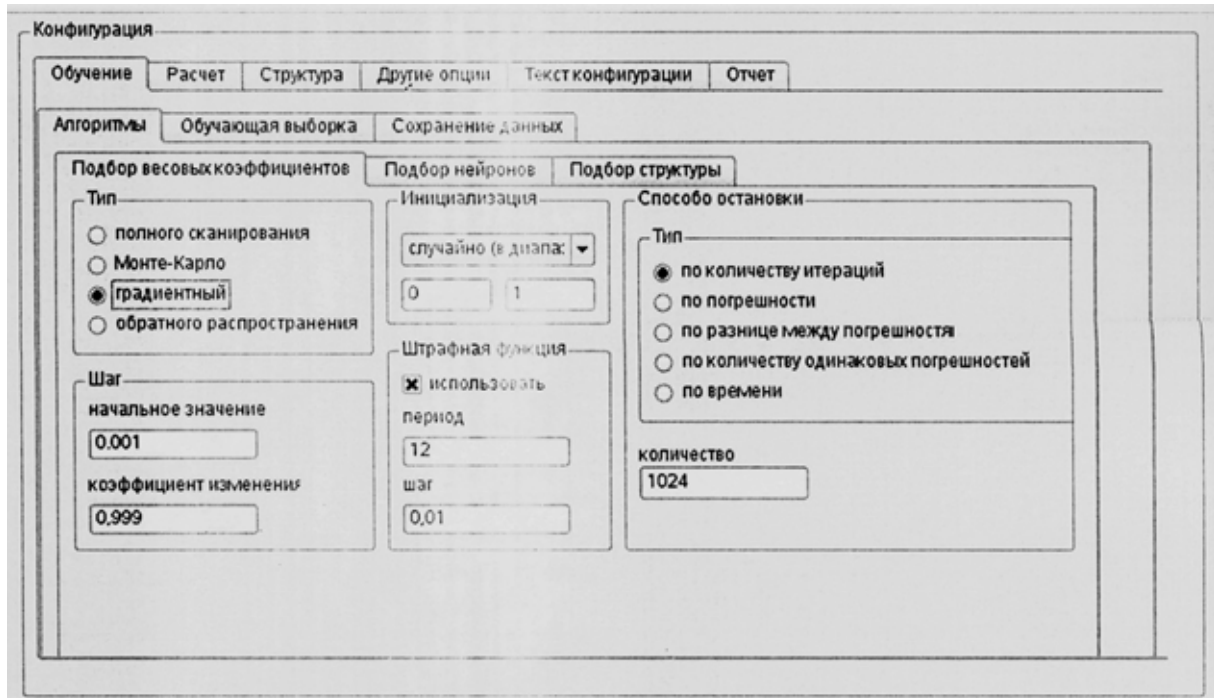


Рис. 4. Интерфейс конфигурации подбора весовых коэффициентов

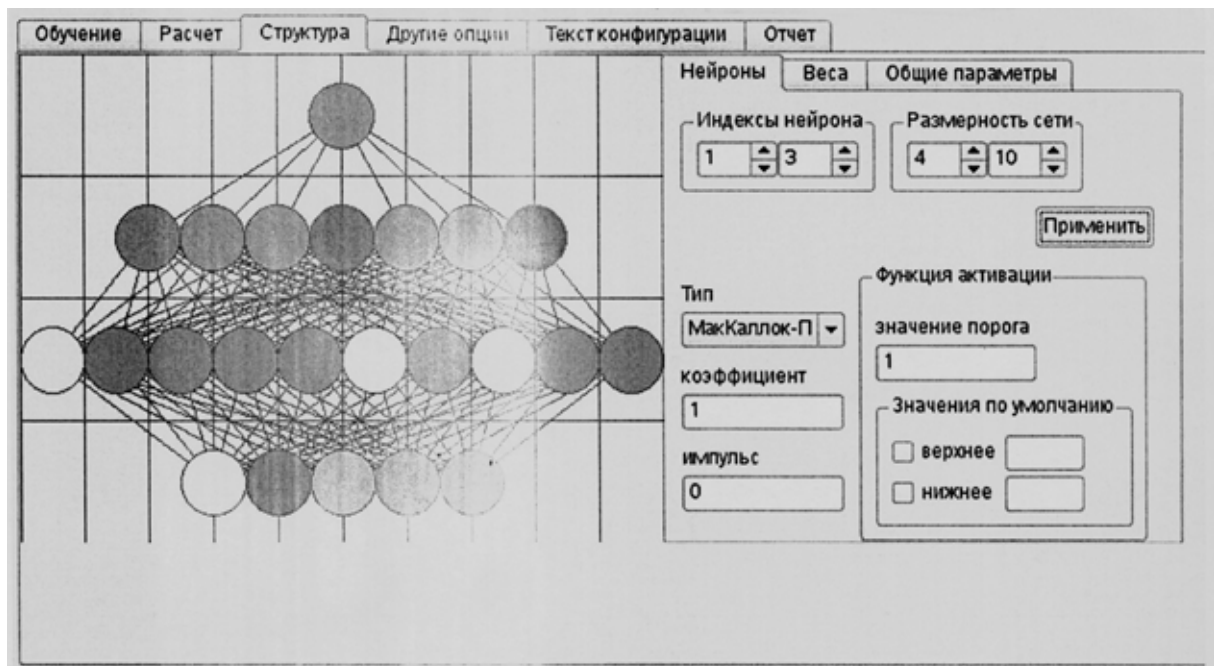


Рис. 5. Конфигурация структуры ИНС

При разработке программы были использованы следующие средства: в качестве IDE была выбрана среда KDevelop. Интерфейс программы был реализован с помощью библиотек QT версии 4.5. Программа может работать под управлением операционных систем GNU/Linux, BSD, MacOS и Microsoft Windows.

Способ ускорения подбора

При полном переборе всех вариантов число состояний μ велико. Его можно выразить формулой

$$l(\mu) = l_N^{\mu}, \quad (5)$$

где l_N — число нейронов в структуре; N_μ — число состояний каждого нейрона. При моделировании сложных объектов структура обычно содержит значительное число нейронов, что, учитывая большое число состояний нейронов N_μ , приводит к тому, что временные затраты на обучение такой сети становятся велики и часто неприемлемы.

Одним из способов ускорения обучения является использование кластерных систем. При этом можно на каждой из систем первый нейрон объявить неизменным и подбирать только остальные. В таком случае число комбинаций вычисляется по формуле $l(\mu) = (l_N - 1)^{\mu}$. Что касается первого нейрона, то на первом вычислительном узле он принимает значение μ_0 , на втором μ_1 и т. д., на

последнем узле μ_n . Ведущий узел устанавливает первые нейроны неизменными и рассылает структуру сети на вычислительные. Каждый узел перебирает все варианты, подбирает оптимальную структуру и возвращает ее на ведущий узел, который устанавливает первый нейрон μ_{n+1} и снова посылает подобранную структуру обратно вычислительному узлу. Такая операция повторяется до тех пор, пока не будут перебраны все состояния первого нейрона. После этого ведущий процессор выбирает минимальное значение невязки.

В качестве кластерной системы может быть использована локальная сеть.

Заключение

Описанный выше алгоритм избавляет от необходимости ручного конфигурирования активационных функций нейронов, а использование кластерных систем приводит к тому, что этот алгоритм не требует излишних временных затрат.

Список литературы

1. Арзамасцев А. А., Крючин О. В., Азарова П. А., Зенкова Н. А. Универсальный программный комплекс для компьютерного моделирования на основе искусственной нейронной сети с самоорганизацией структуры // Вестник Тамб. ун-та. Сер. естеств. и техн. науки. Тамбов, 2006. Т. 11. Вып. 4. С. 564—570.
2. Заенцев И. В. Нейронные сети: основные модели: Учеб. пособие. Воронеж: Изд. ВГУ, 1999. 22 с.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

УДК 004

О. Д. Жуков, канд. техн. наук, вед. науч. сотр.,
Московский государственный университет им. М. В. Ломоносова,
e-mail: zhukovo@dec1.sinp.msu.ru

Информационные технологии на основе полиномиальной алгебры

Одним из путей повышения производительности информационно-вычислительных средств является использование модулярных числовых систем.

Работа посвящена обзорному описанию одного из классов модулярных числовых систем — полиномиальной системы вычетов, полиномиального аналога известной из теории чисел системы вычетов. Многие достоинства системы вычетов являются общими и для этой системы. Основное отличие состоит в том, что деление полиномов в данной системе осуществляется проще и быстрее, чем в системе вычетов. Подобные системы могут успешно использоваться для компьютерных полиномиальных вычислений.

Ключевые слова: система вычетов, полиномиальная алгебра, интерполяционный полином

Введение

В связи с широким использованием в информационно-вычислительных технологиях систем компьютерной алгебры появилась необходимость

в эффективных процедурах манипулирования полиномами помимо целых, вещественных или комплексных цифровых данных. В большинстве систем это достигается использованием обычных

базисов (оснований) полиномиальных полей и манипуляцией с векторами коэффициентов.

В случае точной полиномиальной арифметики часто используют целочисленную модулярную арифметику по достаточно большому простому модулю, чтобы обойти увеличение динамической области для промежуточных результатов и ускорить вычислительные процессы. Выбор большого простого модуля потенциально проблематичен; кроме того, часто требуется второй простой модуль для контроля ошибок. Естественно, такая техника не приемлема для вещественных и комплексных полиномов.

В данной статье на основе зарубежных источников демонстрируются основные идеи, методы и алгоритмы полиномиальной системы вычетов, которая есть аналог известной из теории чисел системы вычетов (СВ). Принципиальными достоинствами СВ являются ее естественный параллелизм и относительная простота расширения динамической области для чисел. Ее основные недостатки состоят в том, что деление в общем случае и операции, связанные с определением знака и величины чисел, не являются модулярными и не могут быть выполнены в обычном для СВ режиме. Для них требуются дополнительные, не модулярные действия. Ситуация усложняется, например, в контексте решения систем линейных уравнений. Подобная ситуация актуальна для полиномиальных систем и вычислений для решения систем полиномиальных уравнений.

В СВ базис взаимно простых целых чисел (модулей) используется для представления целого числа вектором его вычетов по каждому из модулей, соответствующих этим вычетам. Таким образом, целое число X представляется в СВ как

$$X = (x_1, x_2, \dots, x_n),$$

где $x_i \equiv X \pmod{m_i} \equiv |X|_{m_i}$.

Система СВ имеет динамическую область, равную $M = \prod_{i=1}^n m_i$ и определяющую набор из M чисел, представляемых уникально в СВ. Обычно этот набор выбирается как $(0, 1, 2, \dots, M-1)$ либо как $(-K, -K+1, \dots, 0, 1, \dots, K)$, где $M = 2K+1$. Если результаты операций не переполняют динамическую область, тогда сложение, вычитание, умножение могут быть выполнены полностью в рамках СВ, и операции по каждому модулю являются независимыми. Если $X = (x_1, \dots, x_n)$ и $Y = (y_1, \dots, y_n)$, то их сумма, разность и произведение представляются как

$$X + Y = (|x_1 \pm y_1|_{m_1}, \dots, |x_n \pm y_n|_{m_n});$$

$$X \cdot Y = (|x_1 \cdot y_1|_{m_1}, \dots, |x_n \cdot y_n|_{m_n}).$$

Из этих выражений следует, что одноразрядные вычеты операндов X и Y обрабатываются попарно, параллельно и независимо друг от друга (здесь под i -м разрядом подразумевается позиция для вычетов по конкретному модулю m_i).

Все эти принципы можно перенести на полиномиальную алгебру. Отметим, что наряду с названными выше немодулярными операциями достаточно трудоемкими являются и процедуры преобразования чисел из СВ в обычные позиционные системы и обратно. Данные преобразования могут быть выполнены на основе Китайской теоремы о вычетах (КТ), которая обеспечивает доказательство уникальности представления чисел в динамическом диапазоне. КТ показывает, как получить величину $X \in (1, \dots, M-1)$ из его представления в СВ:

$$X \equiv \left| \sum_{i=1}^n M_i \left| M_i^{-1} x_i \right|_{m_i} \right|_M,$$

где $M_i = M/m_i$ и мультипликативные инверсии с индексом i формируются по модулю m_i , соответствующему определенному элементу базиса с индексом i .

Отметим, что основная идея полиномиальной системы вычетов (ПСВ) была представлена, например, в известной книге Кнута [1] и в [2]. В этих источниках описывается сложение, вычитание и умножение и специальный случай целочисленного деления. Кроме того, представлен алгоритм для деления методом "проб" ("trial division"), который основан на знании степеней как делимого, так и делителя. Однако существуют ситуации, когда деление заранее может быть определено как точное, такое, как в случае целочисленного для полиномиальной линейной алгебры [3].

Проверочный шаг метода пробного деления сводится к вычислению и использованию интерполяционного полинома. Это эквивалентно обратному преобразованию на основе КТ в обычную полиномиальную форму. Издержки преобразований становятся незначительными, если они требуются только при выводе результатов. Это позволило бы осуществлять вычислительный процесс исключительно в СВ с высоким темпом модулярных операций.

Очевидно, большой вопрос состоит в возможности развития методов и алгоритмов модулярной алгебры для выполнения деления в общем случае и других важных полиномиальных процедур без необходимости преобразований. В разделе 1 даются основные аспекты в части представления чисел в ПСВ. Они включают не только саму форму представления, но также процедуру восстановления полинома из его представления в ПСВ на основе КТ, которая в этом случае интерпретиру-

ется как интерполяция Лагранжа. Обсуждается выбор подходящего базиса ПСВ. Рассматриваются простейшие полиномиальные операции. Здесь они приводятся для полноты иллюстрации ПСВ.

Раздел 2 посвящен развитию алгоритмов простых операций в ПСВ и начинается с метода точного целочисленного деления, выполняемого непосредственно в ПСВ (без преобразований), если ни один вычет не равен нулю. Если делитель имеет нулевой вычет, то соответствующий вычет числителя также должен не учитываться. Чтобы выполнить деление в таких случаях, требуется расширить базис ПСВ. Эта процедура также рассматривается. Одно из приложений идеи расширения ПСВ есть оценка полинома, которая в нашем случае играет важную роль для неточной операции деления на линейный фактор. Эта операция и деление на несокращаемый квадратичный полином ("квадратик") могут быть выполнены с помощью определения остатка с последующим использованием алгоритма точного деления.

1. Представление в ПСВ

Представление в ПСВ соотносится с хорошо известными идеями полиномиальной интерполяции. В разделе описываются основные принципы и обозначения для такого представления. Также обсуждаются выбор и простейшие операции сложения и умножения.

1.1. Основные принципы и обозначения. Представим полином в кольце $Z(x)$ его вычетами от деления на базисные (моник, линейные и т. п.) полиномы $p_0(x), p_1(x), \dots, p_N(x)$. Этот набор покрывает пространство Π_N полиномов с наибольшей степенью не более N . Пусть

$$p_i(x) = x - x_i. \quad (1)$$

Тогда вычеты полинома $Q(x) \in Z(x)$, относящиеся к данному базису, определяются как

$$q_i \equiv |Q(x)|_{p_i} = Q(x_i). \quad (2)$$

Это представление уникально для $Q \in \Pi_N$. Данное пространство является динамическим диапазоном ПСВ, использующим этот базис или любой базис с $N + 1$ полиномами, удовлетворяющими (1) для различных точек (узлов) x_0, x_1, \dots, x_N .

Пример 1. Даны полиномы степени 2:

$A(x) = x^2 + x + 1$ и $B(x) = 2x^2 + 3$, использующие базис

$$(p_0 = (x + 1), p_1 = x, p_2 = (x - 1)).$$

В соответствии с (1) представлениями $A(x)$ и $B(x)$ в ПСВ являются:

$$A(x) \sim (A(-1), A(0), A(1)) = A = (1, 1, 3);$$

$$B(x) \sim (B(-1), B(0), B(1)) = B = (5, 3, 5).$$

Только полиномы в Π_2 имеют такие представления для заданного базиса.

1.2. Китайская теорема о вычетах – интерполяция Лагранжа. Уникальность представления полиномов в Π_N следует из уникальности полинома интерполяции Лагранжа. Стандартная форма КТ для восстановления полинома Q из его вычетов записывается как

$$Q(x) = \sum_{i=0}^N m_i(x) |_{m_i}^{-1} q_i |_{p_i},$$

где $m_i(x) = \prod_{j \neq i} (x - x_j) = m(x_i)/(x - x_i)$,

Произведение всех элементов базиса $M(x) = \prod_{i=0}^N (x - x_i)$ подобно диапазону M для СВ, равному произведению всех модулей. Отметим, что $|m_i^{-1}|_{p_i} = 1/m_i(x_i)$, так что представление на основе КТ может быть переписано как

$$Q(x) = \sum_{i=0}^N (m_i(x)/m_i(x_i))q_i. \quad (3)$$

Это есть не что иное, как формула интерполяции Лагранжа.

Пример 2. Используя тот же базис примера 1, на основе (3) восстановим исходные полиномы. В данном случае $M_0(x) = x(x - 1)$, $M_1(x) = x^2 - 1$, $M_2(x) = x(x + 1)$ и их обратные представления суть $1/2, -1, 1/2$. На основе (3) получим:

$$A(x) = (1/2)x(x - 1) - (x^2 - 1) + (3/2)x(x + 1) = x^2 + x + 1;$$

$$B(x) = (5/2)x(x - 1) - 3(x^2 - 1) + (5/2)x(x + 1) = 2x^2 + 3,$$

т. е. ожидаемый результат.

1.3. Выбор базиса. Выбор элементов базиса является важным аспектом для конкретной СВ. Во всех случаях базис СВ должен состоять из взаимно простых модулей. Их произведение определяет динамическую область, что накладывает по крайней мере одно ограничение на этот выбор. Оптимальными являются модули вида $2^k, 2^k - 1$, поскольку реализация двоичной модулярной арифметики на их основе является простой. Для модулей общего вида может быть использована модулярная арифметика на основе двоичной технологии, представленной, например, в [1].

Для точного деления в СВ необходимо, чтобы все модули были простыми числами. Такая ситуация возникает, например, при целочисленном решении линейных систем [4]. Если используется комплексная целочисленная арифметика, тогда набор элементов базиса ограничивается простыми

числами Гаусса вида $p = 4k + 1$, так что равенство $x^2 + 1 = 0$ имеет два различных корня в поле Z_p .

При различных узлах x_0, x_1, \dots, x_N базис, определяемый в (1), состоит из простых элементов $Z(x)$ и нет очевидности преимущества, достигаемого конкретным выбором этих узлов. Однако с точки зрения анализа ошибок для интерполяции Лагранжа как метода аппроксимации выбор узлов, конечно, важен. В то же время эти факторы не нарушают правила компьютерной алгебры для полиномов. Для большинства примеров, приведенных ниже, используется простой выбор узлов. Для нечетного числа узлов, соответствующих четному числу $N = 2k$, узлами являются $0, \pm 1, \pm 2, \dots, \pm k$. Для L нечетного к данному набору добавляется узел $k + 1$ или $-k - 1$.

1.4. Сложение и умножение в ПСВ. Сложение или вычитание двух полиномов в ПСВ выполняется простым попарным суммированием или вычитанием их вычетов. Это есть полностью параллельная операция, так как все модули независимы и отсутствуют переносы между разрядами, соответствующими различным модулям. Таким образом, если два полинома $A(x)$ и $B(x)$ представляются в ПСВ как

$$A(x) = (a_1, a_2, \dots, a_N) \\ \text{и } B(x) = (b_1, b_2, \dots, b_N),$$

тогда справедливо

$$A(x) \pm B(x) = \\ = (a_1 \pm b_1, a_2 \pm b_2, \dots, a_N \pm b_N),$$

поскольку $(A \pm B)(x_i) = A(x_i) \pm B(x_i)$.

Подобие с целочисленной арифметикой СВ очевидно.

Пример 3. Для исходных $A(x)$ и $B(x)$ сумма и разность определяются как

$$A(x) + B(x) = (1, 1, 3) + (5, 3, 5) - (6, 4, 8); \\ A(x) - B(x) = (1, 1, 3) - (5, 3, 5) - (-4, -2, -2).$$

Используя (3), получаем

$$(A + B)(x) = (6/2)x(x - 1) - 4(x^2 - 1) + \\ + (8/2)x(x + 1) = 3x^2 + x + 4;$$

$$(A - B)(x) = (-4/2)x(x - 1) - (-2)(x^2 - 1) + \\ + (-2/2)x(x + 1) = -x^2 + x - 2.$$

Отметим, что это преобразование включается здесь только для проверки. Подобным образом умножение может быть выполнено поразрядно и параллельно посредством умножения соответствующих вычетов. При этом результат не должен выходить за динамический диапазон представления чисел в ПСВ. Однако для полиномов нашего примера произведение выходит за этот диапазон. В результате умножения имеем произведение $(1, 1, 3) \cdot (5, 3, 5) = (5, 3, 15)$.

Истинное произведение заданных полиномов есть $2x^4 + 2x^3 + 5x^2 + 3x + 3$, а его значения от $x = (-1, 0, 1)$ суть $(5, 3, 15)$ соответственно. Таким образом, полученное произведение корректно по основанию этого базиса. Отметим, что $(5, 3, 15)$ есть представление в ПСВ полинома $7x^2 + 5x + 3$, который является остатком от деления выше приведенного полинома на $M(x) = x^3 - x$:

$$2x^4 + 2x^3 + 5x^2 + 3x + 3 = \\ = (2x + 2)(x^3 - x) + 7x^2 + 5x + 3.$$

Следовательно, при умножении существует лишь одна проблема расширения динамической области, которая теперь определяется возросшей степенью полинома произведения. В этом случае необходимо выполнить расширение базиса ПСВ. По сравнению с тем, что данная процедура является относительно трудоемкой в СВ, она является достаточно простой для ПСВ. В дальнейшем это будет показано.

2. Простые операции в ПСВ

В этом разделе обсуждаются некоторые вопросы, связанные с базисом и другими полиномиальными операциями, которые легко выполняются в ПСВ.

2.1. Алгоритм точного деления. Деление полиномов в общем случае есть не простая операция. Однако, если нам известно заранее, что делитель является фактором делимого, такое деление может быть выполнено в рамках ПСВ очень просто. Ситуация усложняется при целочисленном решении линейных систем уравнений. Примеры целочисленных алгоритмов для линейных и полиномиальных уравнений описаны в [3].

Точное деление может быть выполнено как обычная модулярная операция в ситуации, где ни один из вычетов делителя не равен нулю.

Пусть $A(x)$, $B(x)$ суть два полинома в ПСВ представляются как $A(x) = (a_1, a_2, \dots, a_L)$ и $B(x) = (b_1, b_2, \dots, b_L)$.

Предположим, что $B(x)$ делит точно $A(x)$. Обозначим полином частного как

$$Q(x) = (q_1, q_2, \dots, q_L).$$

Тогда

- для любого i , для которого $b_i \neq 0$,

$$q_i = a_i/b_i; \quad (4)$$

- если $b_i = 0$, то также $a_i = 0$, поэтому q_i является неопределенным вычетом в соответствии с (4).

Действительно, справедливость первого положения следует из уникальности полинома интерполяции Лагранжа, так как

$$Q(x) = A(x_i)/B(x_i).$$

Второе положение также справедливо, поскольку дано условие $B(x)|A(x)$, при котором $Q(x_i)$ должно остаться конечным, даже если $B(x_i) = 0$.

Единственный остающийся вопрос заключается в том, как вычислить вычеты частного в точках, где $a_i/b_i = 0/0$ являются неопределенными величинами. Но это можно сделать с помощью расширения базиса (основания) ПСВ. Данная процедура использует тот факт, что для любого узла с $b_i = 0$ величина $(x - x_i)$ является фактором A и B , и поэтому частное может быть представлено в ПСВ с базисом $(p_0(x), p_1(x), \dots, p_L(x))/p_i(x)$. Чтобы дополнить представление Q в расширенной ПСВ, необходимо получить вычеты $Q(x_i)$, которые не могут быть определены на основе формулы (4). В этом случае требуется процедура расширения базиса ПСВ от $(p_0(x), p_1(x), \dots, p_L(x))/p_i(x)$ до $(p_0(x), p_1(x), \dots, p_L(x))$. Здесь также можно отметить аналогию с целочисленным сложением для алгебраического решения линейных уравнений [4].

Пример 4. Используя базис ПСВ предыдущего примера, делим $(x^2 - 4)$ на $(x - 2)$.

Теперь $(x^2 - 4) = (-3, -4, -3)$ и $(x - 2) = (-3, -2, -1)$. Отсюда следует, что частное имеет вычеты

$$(-3/-3, -4/-2, -3/-1) = (1, 2, 3),$$

которые, очевидно, являются вычетами корректного результата $x + 2$ при $(-1, 0, 1)$ соответственно. В данном примере все вычеты определяются на основе формулы (4).

Пример 5. В ПСВ предыдущего примера делим $(x^2 - 1)$ на $(x + 1)$. Здесь $(x^2 - 1) = (0, -1, 0)$ и $(x + 1) = (0, 1, 2)$. Первый вычет частного является неопределенным, а остальные вычеты суть $-1, 0$. В ПСВ с базисом $x, x - 1$ модульные основания m_i суть несокращаемые полиномы: $m_1(x) = x - 1$ и $m_2(x) = x$. Определенные выше вычеты представляют выражение для частного в сокращенном базисе как

$$(-1/m_1(0))m_1(x) + (0/m_2(1))m_2(x) = x - 1.$$

После расширения базиса получим недостающий вычет $q_0 = (x - 1) = -2$ при $x = -1$. Таким образом, результатом деления является $Q = (-2, -1, 0)$. Этот код соответствует совокупности вычетов для $(x - 1)$ в узле $-1, 0, 1$.

2.2. Процедура расширения ПСВ. Обычный способ расширения базиса в ПСВ осуществляется с помощью представления в системе со смешанными основаниями (ССО), в которой цифры целого X не являются коэффициентами при возрастающих степенях фиксированного основания. Вместо этого они являются коэффициентами при следующих произведениях различных оснований:

$$1, p_1, p_1 p_2, \dots, p_1 p_2 \dots p_{L-1}.$$

Данные коэффициенты могут быть восстановлены непосредственно из представления в СВ с помощью простой рекурсии. Целое X может быть записано в ССО как

$$X = x_1 + x_2 p_1 + x_3 p_1 p_2 + \dots + x_L p_1 p_2 \dots p_{L-1}.$$

Очевидно, справедливо, что

$$a_1 = x_1 \text{ и } a_{k+1} \equiv |P_k^{-1}(X - S_k)|_{p_{k+1}},$$

где $P_k = \prod_{j=1}^k p_j$, $S_k = \sum_{j=1}^k a_j P_{j-1}$, так что S_k суть частные суммы приведенного выше представления в ССО. Отметим, что само по себе знание X не требуется, в то время как все операции здесь модулярные, и поэтому соответствующие вычеты могут быть использованы в рамках процедуры рекурсии; значения $|P_j|_{p_{k+1}}$ и $|P_k^{-1}|_{p_{k+1}}$ могут быть вычислены и запомнены заранее.

Процесс определения вычетов полинома в ССО на основе его представления в СВ состоит в том, чтобы выразить $Q(x) = (q_0, q_1, \dots, q_L)$ в форме

$$Q(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \dots + (x - x_0)(x - x_1)\dots(x - x_{L-1})a_L.$$

Сравнение данного выражения с формулой разделенных разностей Ньютона показывает, что коэффициенты этого выражения суть разделенные разности:

$$a_k = Q[x_0, x_1, \dots, x_k].$$

Отметим, что формула разделенных разностей очень эффективна для преобразования в ССО и, следовательно, может быть использована для вычисления интерполяционного полинома.

Однако преобразование в ССО не является необходимым для расширения базиса ПСВ. Мы можем использовать формулу (3) для оценки новых узлов. Новые узлы включают те, в которых вычеты являются неопределенными при делении. Данная процедура сводится к простым численным операциям, которые могут быть выполнены параллельно. Эта простота объясняется тем, что вычеты по модулям, равным линейным полиномам, соответствуют оценкам в узлах. С учетом формулы (3) должны быть запомнены полиномы $m_i(x)$ в факторизованной форме и оценки $m_i(x_i)$. Оценка в новом узле $Q(x_{L+1})$ может быть, таким образом, получена с помощью простых арифметических операций. В рамках компьютерных технологий оценки $m_i(x_{L+1})$ могут быть получены одновременно (параллельно). Если $m_i(x)$ запомнены в факторизованной форме, то возможна модификация метода следующим образом. Факторы для неопределенных узлов должны быть просто опу-

щены с последующей соответствующей модификацией значений $m_i(x_i)$.

Отметим простоту расширения наборов $m_i(x)$, $m_i(x_i)$ после расширения базиса посредством умножения каждого из них на соответствующие дополнительные факторы. Исходный полином

$$M = \prod_{i=1}^L (x - x_i)$$

становится полиномом $m_{L+1}(x)$,

и M также увеличивается на значение нового фактора. Рассмотрим расширение базиса на конкретном примере.

Пример 6. Требуется расширить ПСВ из примера 1 путем добавления двух новых базисных элементов $(x \pm 2)$ и затем умножить два заданных в исходной ПСВ полинома в расширенной ПСВ.

Для заданной ПСВ

$$m_0(x) = x(x-1); m_1(x) = x^2 - 1; m_2(x) = x(x+1);$$

$$m_i(x_i) = 2, -1, 2. \text{ Также } A(x) = x^2 + x + 1 = (1, 1, 3).$$

Отсюда следует

$$a_3 = |A|_{x+2} = A(-2) = (1/2)(-2)(-3) + (1/-1)(3) + (3/2)(-2)(-1) = 3.$$

Аналогично $a_4 = |A|_{x-2} = 7$. Применяя тот же алгоритм к $B(x) = 2x^2 + 3$, получим $b_3 = b_4 = 11$. Таким образом, в расширенной ПСВ с узлами $-1, 0, 1, -2, 2$ имеем

$$A = (1, 1, 3, 3, 7); B = (5, 3, 5, 11, 11).$$

Их произведение имеет представление в расширенной ПСВ

$$AB = (5, 3, 15, 33, 77).$$

Мы можем проверить корректность этих вычетов для произведения $A(x)B(x) = 2x^4 + 2x^3 + 5x^2 + 3x + 2$ во всех узлах. Для восстановления полинома на основе теоремы КТ и интерполяции Лагранжа нам необходим расширенный набор

$$m'_i: m_0(x) = x(x-1)(x^2-4);$$

$$m_1(x) = (x^2-1)(x^2-4);$$

$$m_2(x) = x(x+1)(x^2-4);$$

$$m_3(x) = x(x^2-1)(x-2);$$

$$m_4(x) = x(x^2-1)(x+2).$$

Значения $m_i(x_i)$ тогда суть $-6, 4, -6, 24, 24$.

Однако это преобразование обычного полинома на основе КТ не является необходимым, пока окончательный результат не потребует пользователя.

Пример 7. Полином, чье представление в ПСВ, соответствующее узлам $-2, -1; 0, 1, 2$, есть $A = (0, -3, -4, -9, 0)$, точно делится на $B(x) = (0, -3, -4, -3, 0)$. Требуется найти частное в ПСВ.

Ясно, что первый и последний вычеты частного являются неопределенными для модулярного деления. Три остальные вычеты суть $1, 1, 3$ в узлах $-1, 0, 1$. Эти узлы дают нам $m_1(x) = x(x-1)$; $m_2(x) = x^2 - 1$; $m_3(x) = x(x+1)$. Они суть узлы и полиномы, используемые выше, с измененными лишь индексами для отображения определенного здесь порядка следования узлов. Соответствующие $m_i(x_i)$ суть $2; -1; 2$. Тогда мы получим

$$q_0 = (1/2)m_1(-2) + (1/-1)m_2(-2) + (3/2)m_3(-2) = 3;$$

$$q_4 = (1/2)m_1(2) + (1/-1)m_2(2) + (3/2)m_3(2) = 7,$$

так что частное представляется как $Q = (3, 1, 1, 3, 7)$.

Действительно, так как $(x^4 + x^3 - 3x^2 - 4x - 4)/(x^2 - 4) = x^2 + x + 1$, то легко проверить корректность остатков, приведенных выше.

2.3. Оценки полиномов. Процедура полиномиальных оценок в ПСВ выполняется просто с помощью только что описанного метода расширения базиса. Вычет полинома Q по модулю, соответствующему нормированному линейному полиному $x - x_k$, дается как $Q(x_k)$ в (2). Таким образом, задача расширения базиса для включения нового полинома $x - x_k$ в точности равноценна оценке полинома в узле x_k . Единственную дополнительную информацию составляют значения базисных полиномов Лагранжа $m_i(x_k)$ в конкретном узле. Эти значения могут быть легко вычислены на параллельной процессорной системе.

Алгоритм оценки полинома в ПСВ:

- Ввод: базисные узлы ПСВ x_0, x_1, \dots, x_L и значения $m_i(x_i)$ для этих узлов.
- Представление в ПСВ $Q = (q_0, q_1, \dots, q_L)$.
- Оценка для узла x .
- Инициализация:

$$\mu_i := q_i/m_i(x_i)$$

для $i = 0, 1, \dots, L$ (параллельно).

Вычислить: для $i = 0 : L$ (параллельно); для $j = 0 : L$

$$\mu_j := \mu_i(\bar{x} - x_j) \text{ для } j \neq i.$$

- Вывод: $Q(\bar{x}) = \sum_{i=0}^L \mu_i$

Пример 8. Требуется определить оценку $Q(1.5)$ для полинома, чье представление в ПСВ, соответствующее узлам $-2, -1, 0, 1, 2$, есть $(3, 1, 1, 3, 7)$.

Здесь ввод для нашего алгоритма состоит из узлов, оценок $m_i(x_i) = 24, -6, 4, -6, 24$ и вычетов. Инициализация, таким образом, выполняется с $\mu_i = 1/8, -1/6, 1/4, -1/2, 7/24$ или $1/24, (3, -4, 6, -12, 7)$. Значения $(\bar{x} - x_j)$ суть $3.5, 2.5, 1.5, 0.5, -0.5$,

так что после многократного выполнения стадий j -го цикла модифицированные значения суть:

$$\begin{aligned} &(1/48)(6, -28, 42, -84, 49), \\ &(1/96)(30, -56, 210, -420, 245), \\ &(1/192)(90, 168, 420, -1260, 735), \\ &(1/384)(90, -168, 420, -2520, 735), \\ &(1/768)(-90, 168, -420, 2520, 1470). \end{aligned}$$

Суммируя эти окончательные входные значения, получим величину $19/4$, что соответствует именно $x^2 + x + 1$ с оценкой в узле 1,5 (см. пример 7). Определение данной оценки показано детально, чтобы проиллюстрировать возможность параллелизма проводимых вычислений. Процедура обеспечивает расширение базиса ПСВ в целях включения узла 1,5.

Эта же идея, которая обеспечивает эффективный способ расширения базиса и оценки, является основой для неточного полиномиального деления.

2.4. Деление на линейные полиномы. Простейший случай точного деления полиномов есть деление на линейный фактор. Алгоритм для этого случая описывается просто:

- найти вычет, который соответствует оценке;
- вычесть этот вычет из делимого;
- использовать алгоритм точного деления.

Если делитель не является нормированным полиномом, может быть использовано предварительное масштабирование числителя и знаменателя, чтобы нормировать делитель в виде $(x - \bar{x})$. Также мы можем считать делитель как полином вида $a(x - \bar{x})$. Теперь остаток от деления полинома $Q(x)$ на $(x - \bar{x})$ есть $Q(\bar{x})$. Определение этого остатка и соответствует оценке полинома. Алгоритм из раздела 2.3 может быть использован для вычисления вычета.

Разность $Q(x) - Q(\bar{x})$ делится без остатка на $(x - \bar{x})$ и поэтому может быть использован алгоритм точного деления из раздела 2.1. Частное тогда равно $(Q(x) - Q(\bar{x})) / (a(x - \bar{x}))$, а остаток — $Q(\bar{x})$. При этом делитель есть линейный фактор делимого, и алгоритм завершается точным делением.

Пример 9. Найти частное и вычет (остаток) после деления $x^2 + x + 1$ на $2x - 3$ с использованием ПСВ с узлами $-2, -1, 0, 1, 2$.

Сначала запишем процедуру как

$$(x^2 + x + 1) / (2x - 3) = (x^2 + x + 1) / 2(x - (3/2)).$$

Мы уже знаем (см. пример 7), что вычеты для полинома $x^2 + x + 1$ в этих узлах суть $(3; 1; 1; 3; 7)$. Также в примере 8 мы сделали оценку $x^2 + x + 1$ при $x = 3/2$. С помощью дедукции мы можем установить, что остаток для этой операции равен $19/4$. Вычитание этого остатка из числителя экви-

валентно вычитанию его из каждого вычета. Таким образом, для заданного базиса ПСВ

$$\begin{aligned} &(x^2 + x + 1) - 19/4 = \\ &(-7/4, -15/4, -15/4, -7/4, 9/4). \end{aligned}$$

Вычеты делителя суть

$$(-7, -5, -3, -1, 1).$$

Поэтому вычеты частного суть

$$(1/4, 3/4, 5/4, 7/4, 9/4),$$

которые идентифицируются как вычеты полинома

$$(1/2)x + 5/4 = (1/4)(2x + 5).$$

Проверка подтверждает, что

$$x^2 + x + 1 = (1/4)(2x + 5)(2x - 3) + 19/4.$$

Отметим, что этот алгоритм опирается на знание того, что делитель является линейным полиномом. Если делитель представлен в ПСВ и не известно, является ли он линейным, тогда используется общий алгоритм деления из раздела 3. Первый шаг этого алгоритма состоит в определении степени делителя. Затем может быть использован приведенный выше метод.

2.5. Деление на квадратичные полиномы. Последний специальный случай деления полиномов в ПСВ относится к делению на квадратичные полиномы, которые для краткости будем называть квадратиксами. Сначала рассмотрим ситуацию, когда делитель является сокращаемым квадратиксом и имеет отличные линейные факторы.

2.5.1. Случай сокращаемого квадратичного делителя. Предположим, что требуется разделить полином $A = (a_0, a_1, \dots, a_L)$ на сокращаемый (приводимый) квадратикс $B = (b_0, b_1, \dots, b_L)$, представленный в ПСВ с узлами

$$x_0, x_1, \dots, x_L.$$

Из любых трех вычетов полинома B можно определить его линейные факторы $(x - v)$, $(x - w)$ при $v \neq w$ (специальный случай с $v = w$ может быть проверен повторным делением на линейный фактор). Обозначив частное и остаток от деления соответственно как Q и R , получим

$$A(x) = Q(x)B(x) + R(x).$$

Из этого следует

$$r_v := R(v) = A(v), \quad r_w := R(w) = A(w). \quad (5)$$

Эти величины могут быть получены при использовании алгоритма оценки из раздела 2.3 таким же образом, как и для линейного делителя. Из цифр остатка может быть определено представление R в ПСВ:

$$r_i = R(x_i) = ((x_i - v)r_w - (x_i - w)r_v) / (w - v). \quad (6)$$

Это есть простой случай интерполяции Лагранжа. Полином $A - R := (a_0 - r_0, a_1 - r_1, \dots, a_L - r_L)$ кратен делителю B , и поэтому здесь можно применить алгоритм точного деления из раздела 2.1. Отметим, что как определение вычетов в (5), так и последующее определение представления R в ПСВ являются полностью распараллеливаемыми процедурами.

Пример 10. Требуется разделить полином $A = (0, -3, -4, -9, 0)$ на квадратикс $B = (5, 0, -3, -4, -3)$, где представление в ПСВ использует узлы $-2, -1, 0, 1, 2$.

В этом случае легко проверить, что вычеты B суть вычеты полинома $x^2 - 2x - 3 = (x + 1)(x - 3)$. Вычеты на основе этих двух корней определяются как

$$r_v = R(-1) = A(-1) = -3;$$

$$r_w = R(3) = A(3) = 65.$$

На основе (6) вычислим вычеты остатка R в заданных узлах:

$$r_i = (65(x_i + 1) + 3(x_i - 3))/4,$$

что соответствует $R := (-20, -3, 14, 31, 48)$. Вычитая R из A , получим разность

$$A - R := (20, 0, -18, -40, -48),$$

которая делится на B без остатка. Вычет частного в узле -1 является неопределенным в результате деления:

$$Q := (20/5, 0/0, -18/-3, -40/-4, -48/-3) = (4, ?, 6, 10, 16).$$

Неопределенный вычет определяется на основе алгоритма расширения ПСВ: $q_2 = 4$. Таким образом, частное есть $Q := (4, 4, 6, 10, 16)$.

Рассмотрим теперь случай деления на сокращаемый делитель, который есть квадрат линейного фактора: $B = (x - v)^2$. Принцип остается неизменным. Однако имеются отличия в деталях.

Как обычно, представим $A(x) = Q(x)(x - v)^2 + R(x)$. Оценка в узле v дает $R(v) = A(v)$. Обозначив $R(x) = m(x - v) + c$, получим $c = A(v)$. Из этого следует, что $A(x) - c$ делится без остатка на $(x - v)$ и

$$(A(x) - c)/(x - v) = Q(x)(x - v) + m.$$

Таким образом, c есть остаток и выражение $A(x)' = (A(x) - c)/(x - v)$ является частным, если A нацело делится на линейный фактор $(x - v)$. Аналогично m есть остаток и Q является частным от деления A' на этот же линейный фактор. Вся процедура сводится, таким образом, к повторному делению на линейный фактор. Алгоритм точного деления дает представление Q в ПСВ. В то же время вычеты R определяются на основе соотношения $R(x_i) = m(x_i - v) + c$.

2.5.2. Несокращаемые делители. Те же основные принципы, которые были использованы для ли-

нейных и сокращаемых квадратиксов, могут быть применены для несокращаемых делителей. Метод сводится к следующим действиям:

- найти корни делителя;
- вычислить остаток и вычесть его;
- использовать алгоритм точного деления.

Существенное отличие состоит в том, что поскольку корни в этом случае суть комплексные числа, оценки выполняются в комплексной арифметике. Однако для получения всей необходимой информации достаточна оценка числителя в одной из пар комплексных сопряженных корней. Это в некоторой степени упрощает процедуру. Остальные операции выполняются исключительно в поле вещественных чисел. Здесь, как и прежде, большой объем вычислений характеризуется естественным параллелизмом.

Предположим теперь, что требуется разделить полином $A := (a_0, a_1, \dots, a_L)$ на несокращаемый квадратикс $B := (b_0, b_1, \dots, b_L)$, где оба полинома представлены в ПСВ при использовании узлов x_0, x_1, \dots, x_L . На основе любых трех остатков полинома B могут быть определены его комплексные линейные факторы, например, $(x - v)$, $(x - \bar{v})$. Имеем

$$A(x) = Q(x)B(x) + R(x),$$

где известно, что $R(x)$ — линейный полином. Оценка $A(v)$ может быть использована для определения вычетов остатка $R(x)$:

$$r_v = R(v) = A(v); r_0 = R(\bar{v}) = \overline{A(v)} = A(v) = \bar{r}_v.$$

Если теперь записать $R(x) = mx + c$, тогда в узлах v, \bar{v} получим.

$$mv + c = A(v)m\bar{v} + c = \overline{A(v)}.$$

Разность этих выражений дает

$$m = \text{Im}(A(v))/\text{Im}(v).$$

Тогда

$$c = \text{Re}(A(v)) - m\text{Re}(v).$$

Вычеты остатка в виде полинома $R(x)$ представляются как

$$r_i = mx_i + c.$$

После вычитания этого остатка из A в ПСВ можно выполнить точное деление и определить частное в виде полинома $Q(x)$.

Пример 11. В ПСВ с узлами $-2, -1, 0, 1, 2$ разделить полином $A := (33, 5, 1, 3, 1, 17)$ на несокращаемый квадратикс $B := (3, 1, 1, 3, 7)$. В этом случае делитель есть $x^2 + x + 1$ (см. пример 7), который имеет комплексные корни $(-1/2)(1 \pm i\sqrt{3})$. Применяя алгоритм оценки с помощью комплексной арифметики для $v = (-1/2)(1 + i\sqrt{3})$, получим $A(v) = (-3/2) + i\sqrt{3}/2$. Это, в свою очередь, дает $m = (\sqrt{3}/2)/(-\sqrt{3}/2) = -1$ и $c = (-3/2) -$

$-(-1)(-1/2) = -2$. Остаток теперь может быть представлен как $R := (0, -1, -2, -3, -4)$. Частное тогда будет равно

$$Q = (A - R)/B := ((33, 5, 1, 3, 17) - (0, -1, -2, -3, -4))/(3, 1, 1, 3, 7) = (11, 6, 3, 2, 3).$$

Действительно, Q представляет полином $x^2 - 2x + 3$, а $R(x) = -x - 2$. Числитель равен $x^4 - x^3 + 2x^2 + 1$ и, таким образом,

$$x^4 - x^3 + 2x^2 + 1 = (x^2 - 2x + 3)(x^2 + x + 1) - x - 2.$$

2.5.3. Деление в общем случае. Выше были рассмотрены специальные случаи деления полиномов в ПСВ, которое для них выполнялось во многом аналогичным способом.

Такой подход не ограничивается этими специальными случаями, если имеется возможность найти остаток. Важным аспектом этого шага в предыдущих случаях являлось то, что была известна степень полинома делителя и, таким образом, полинома остатка. В общем случае деления в ПСВ эта информация сразу недоступна. Тем не менее, общий метод деления в ПСВ может использовать те же основные шаги. Но первый шаг заключается в определении степени делителя, которое требует объемного разъединения, и его рассмотрение выходит за рамки данной работы.

Заключение

В заключение отметим следующее. Приведенные выше принципы представления полиномов и вычислений в ПСВ обладают очевидными достоинствами, например, по сравнению с обычными

полиномиальными системами при использовании параллельной организации вычислений. Особенно потенциальные преимущества ПСВ проявляются для всех арифметических операций.

Результаты сравнения вычислительной сложности методов и алгоритмов для обычных полиномиальных систем и ПСВ сводятся к следующему. Сложность сложения (вычитания), умножения, точного деления, деления на линейный и квадратичный делители для обычных полиномиальных систем определяется соответственно следующими примерными значениями: N , $(N + 1)^2/2$, $(N + 1)^2$, $6N$, $7N$. Те же сложности для ПСВ при последовательной организации вычислений пропорциональны значениям: N , N , N , N^2 , N^2 , а при параллельной обработке — значениям 1 , 1 , 1 , N , N .

Отсюда видно, что при выполнении простых операций ПСВ имеет относительно обычных полиномиальных систем преимущества, аналогичные преимуществам системы вычетов относительно обычной двоичной арифметики. Из этого следует, что ПСВ может служить эффективной основой для высокопроизводительных полиномиальных вычислений в информационно-вычислительных системах определенного класса.

Список литературы

1. Knuth D. E. The art of computer programming. 1981. Vol. 2. Seminumerical algorithms, 2-nd ed., MA.
2. Geddes K. O. Algorithms for computer algebra, Kluwer Acad. Publisher Dordrecht. 1992.
3. Nakos C. C. Fraction-free algorithms for linear and polynomial equations, ACM SIGSAM Bull. 31. 1997. P. 11–19.
4. Turner P. R. Fraction-free RNS algorithms for solving linear systems Washington DC. Proc. ARITH13. 1997. P. 218–224.

УДК 622:519.9

Д. В. Калитин, канд. техн. наук, доц.,

Московский государственный горный университет, e-mail: bryce@rambler.ru

Проектирование параллельных декомпозиций управляющих автоматов в k -значных логиках

Рассмотрена стратегия проектирования параллельной декомпозиции автоматов, основанная на теоретико-графовом вложении графа сцепления автомата в граф специального вида — кластер функциональной несвязности. В результате такой декомпозиции можно спроектировать функции, которые будут описывать работу этих автоматов. Далее полученные функции реализуются с помощью цифровых трехзначных нейронов. Представленная стратегия избавлена от недостатков методов, использовавшихся ранее. Также предложенный метод позволяет получать автоматы-сомножители сразу, т. е. параллельно, а не последовательно, как ранее.

Ключевые слова: конечный автомат, декомпозиция, граф

Повышение производительности и безотказности цифровых бортовых систем, систем жизненного обеспечения горного, морского, космического базирования и других информационно-

вычислительных комплексов определяет необходимость автоматизированного проектирования параллельно функционирующих управляющих автоматов с ослабленной (в пределе минималь-

ной) связностью. Эта проблема, как проблема проектирования параллельной декомпозиции управляющих автоматов, впервые была поставлена Дж. Хартманисом (1961 г.), для решения которой им был предложен специальный алгебраический аппарат разбиений внутренних состояний автомата (нетерминальных символов) со свойством подстановки (СП-разбиений). Развитие этого аппарата (совместно с Р. Стирнсом) привело к созданию алгебры пар, явившейся математической основой результатов, связанных с построением параллельной декомпозиции автоматов. Недостатками этого подхода являются комбинаторный перебор возможных разбиений и малая доля автоматов, обладающих свойством подстановки. Процедура, основанная на алгебре пар, по существу, является процедурой не проектирования, а распознавания свойства подстановки автомата, которая, быть может, только на последнем шаге преобразования даст отрицательный результат, т. е. что автомат не обладает свойством подстановки и не может быть построен.

Синтез параллельной декомпозиции автомата, описываемого графом переходов

$$G_{\text{пер}} = \langle V, (U, (X, Y)) \rangle,$$

где V — множество нетерминальных символов; X и Y — терминальные, соответственно входные и выходные символы; $U, U \subset V^2$ — множество переходов, сводится к поиску частичного декартова произведения $\prod_i G_{\text{пер}i}$, включающего заданный граф переходов:

$$G_{\text{пер}} \subset \prod_i G_{\text{пер}i} \quad (1)$$

При алгебраическом подходе разрабатывались синтаксические процедуры решения задачи декомпозиции, которые при реальном проектировании были не применимы, так как практические управляющие автоматы являются мультипликативно-повторными, т. е. хотя бы два автомата-сомножителя описывались автоматными операторами, имеющими общий аргумент. Синтаксические процедуры при алгебраическом подходе были по существу не процедурами проектирования, а процедурами распознавания мультипликативной бесповторности синтезируемой декомпозиции.

В 1970 г. проф. В. А. Горбатов ослабил требования к решению этой проблемы — искать не необходимые и достаточные условия, а только достаточные, но при этом усилил проблему — строить параллельную декомпозицию автоматом не с *ослабленной функциональной связностью* элементов памяти, а с *минимальной функциональной связностью* параллельно функционирующих автоматов-сомножителей, где под функциональной связно-

стью $L(G_{\text{пер}})$ нетерминальных символов понимается

$$L(G_{\text{пер}}) = \sum_{i,j,k} \alpha_{ik} v_j, G_{\text{пер}} \subset \prod_i G_{\text{пер}i} \quad (2)$$

где α_{ik} — число переходов, в которых при вычислении i -го состояния k -го автомата-сомножителя используется $v_j, j \neq i$, состояний других автоматов-сомножителей.

В том же году проф. В. А. Горбатов решил эту проблему, которую затем развил его ученик проф. А. Г. Дедегкаев (1993 г.).

Конструктивно решение этой проблемы представляло собой последовательную стратегию построения графов переходов автоматов-сомножителей на основе многокомпонентной раскраски графов сцепления. При этом для проектирования *минимально-связной параллельной декомпозиции* необходимо построение дерева всевозможных многокомпонентных раскрасок графов сцепления, что является трудоемкой процедурой.

Для минимизации функциональной связности $L(G_{\text{пер}})$ будем использовать понятие бинарного отношения сцепления $R_{\text{сц}}$ в множестве нетерминальных символов $\{s_i/s_i$ — нетерминальный символ автоматной грамматики}.

Графом сцепления $G_{\text{сц}} = \langle V, (U, \{X_i\}) \rangle$ называется взвешенный граф, элементы носителя V которого взаимно однозначно соответствуют элементам множества нетерминальных символов $v_i \in V, v_i \leftrightarrow s_i$ и две вершины v_i, v_j соединены ребром, взвешенным входным терминальным символом X_k , если из внутреннего состояния s_i под воздействием вектора X_k автомат переходит в состоя-

ние $s_\alpha, s_i \xrightarrow{X_k} s_\alpha$ и из состояния s_j под воздействием вектора $\tilde{X}_k, X_k \subset \tilde{X}_k$, — в состояние $s_\beta, s_j \xrightarrow{\tilde{X}_k} s_\beta$, при этом $s_\alpha \neq s_\beta$ и $(s_i \neq s_\alpha) \& (s_j \neq s_\beta)$:

$$(s_i \xrightarrow{X_k} s_\alpha, s_j \xrightarrow{\tilde{X}_k} s_\beta, s_\alpha \neq s_\beta, (s_i \neq s_\alpha) \vee (s_j \neq s_\beta)) \rightarrow \{s_i, s_j\} \in R_{\text{сц}} \quad (3)$$

Состояния s_i, s_j , удовлетворяющие (3), называются сцепленными вектором X_k .

При представлении графа переходов $G_{\text{пер}} = \langle V, (U, (X, Y)) \rangle$ заданной автоматной грамматики в виде частичного декартова произведения $\prod_{i=1}^n G_{\text{пер}i}$ $G_{\text{пер}i} = \langle V_i, (U_i, (X, Y)) \rangle$, каждому нетерминальному символу $s_a, s_a \leftrightarrow v_a, v_a \in V$ исходной грамматики сопоставляется кортеж $(s_{ai}/i = 1, 2, \dots, n, s_{ai} \leftrightarrow v_{ai}, v_{ai} \in V_i)$. Очевидно, что име-

ем нарушение автоматности l -го графа переходов $G_{перl} = \langle V_l, (U_l, (X, Y)) \rangle, l \in \{1, 2, \dots, n\}$, если при переходе из сцепленных векторами $X_k, \tilde{X}_k, X_k \subset \tilde{X}_k$ состояний

$$\{s_a, s_b\} \in R_{сц}. \quad (4)$$

l -й автомат-сомножитель, имеющий одинаковые значения для нетерминальных символов s_a и s_b и различные значения для состояний s_α и s_β , под воздействием векторов X_k и \tilde{X}_k переходит соответственно в состояния s_a и s_b :

$$s_a \xrightarrow{X_k} s_\alpha, s_b \xrightarrow{\tilde{X}_k} s_\beta, \quad (5)$$

$$s_a \leftrightarrow (s_{a1}, s_{a2}, \dots, s_{al}, \dots, s_{an}),$$

$$s_b \leftrightarrow (s_{b1}, s_{b2}, \dots, s_{bl}, \dots, s_{bn}),$$

$$s_\alpha \leftrightarrow (s_{\alpha1}, s_{\alpha2}, \dots, s_{\alpha l}, \dots, s_{\alpha n}),$$

$$s_\beta \leftrightarrow (s_{\beta1}, s_{\beta2}, \dots, s_{\beta l}, \dots, s_{\beta n}),$$

$$s_{al} = s_{bl} \quad (6)$$

$$s_{\alpha l} \neq s_{\beta l} \quad (7)$$

Отсюда следуют теоремы 1, 2.

Теорема 1.

Если для любой пары кортежей $(s_{ai}/i = 1, 2, \dots, n)$ и $(s_{bi}/i = 1, 2, \dots, n)$, соответствующих сцепленным нетерминальным символам s_a и $s_b, \{s_a, s_b\} \in R_{сц}$ исходной грамматики не найдется разряда, в котором бы кортежи совпадали, то функциональная связность $L(G_{пер})$ параллельной декомпозиции автоматной грамматики равна нулю.

В этом случае не выполняется соотношение (6).

Теорема 2.

Если для пары кортежей $(s_{ai}/i = 1, 2, \dots, n)$ и $(s_{bi}/i = 1, 2, \dots, n)$, соответствующих сцепленным нетерминальным символам s_a и $s_b, \{s_a, s_b\} \in$

$R_{сц}, s_a \xrightarrow{X_k} s_\alpha, s_b \xrightarrow{\tilde{X}_k} s_\beta$, найдется l -й автомат-сомножитель, который переходит в одно и то же состояние при переходах из s_a при терминальном символе X_k и из s_b — при $\tilde{X}_k, X_k \subset \tilde{X}_k$, то нарушение автоматности в l -м автомате-сомножителе не происходит.

В этом случае не выполняется соотношение (7).

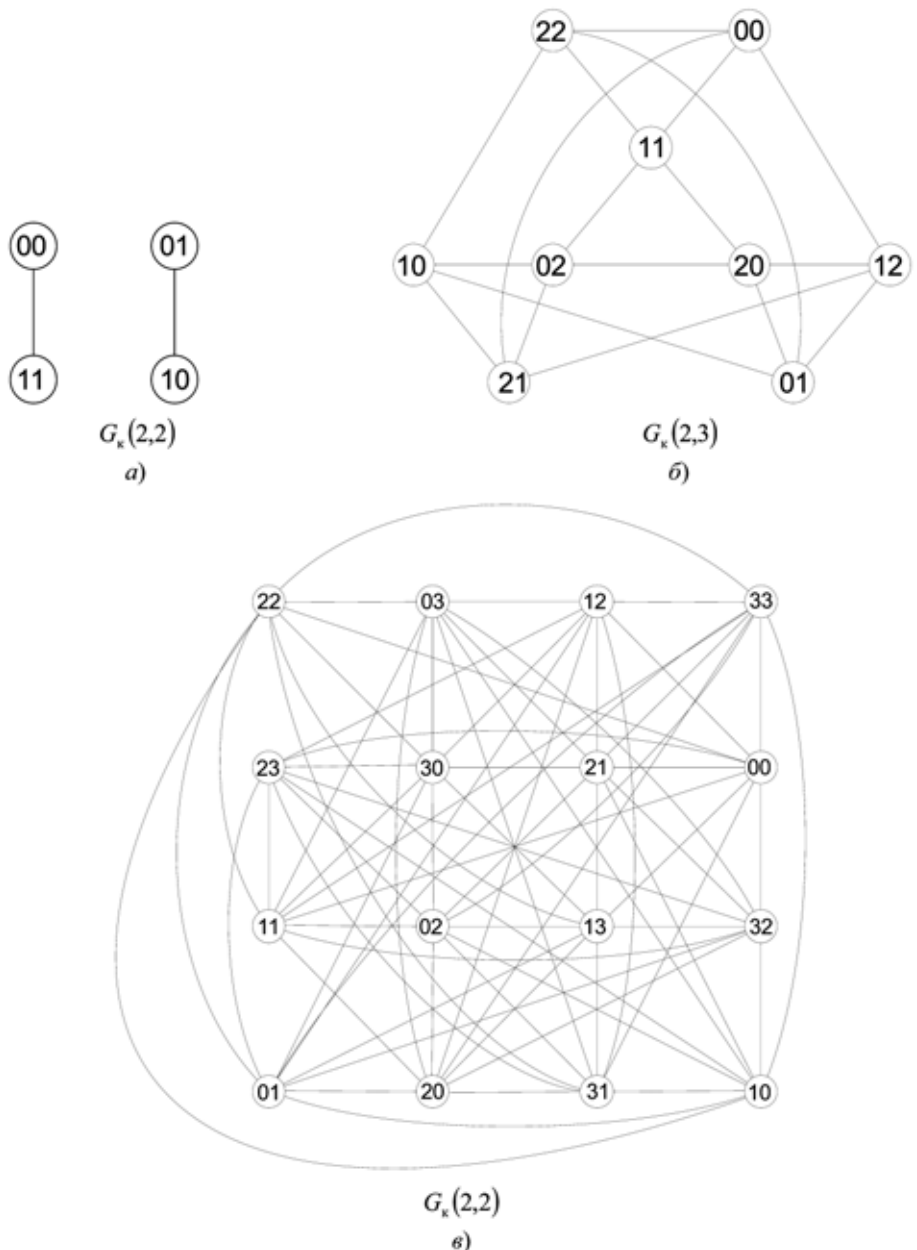


Рис. 1. Вид кластеров в двумерном пространстве для 2-, 3-, 4-значной логики

На основании теоремы 1 синтез параллельной декомпозиции управляющих автоматов сведен к вложению графа сцепления $G_{сц}$ исходной автоматной грамматики в специальный граф $G_k(n, k)$ — кластер, однозначно определяемый значностью k применяемой логики и ее размерностью n соответствующего пространства $P(n, k)$. Рассмотрим это пространство $P(n, k)$.

Построим кластер G_k , состоящий из точек $\{Z_i\}$ этого пространства, каждая пара которых $\{Z_i, Z_j\}$ соединены ребром, если векторы Z_i, Z_j имеют различные значения в каждом разряде $l, z_{ij} \neq z_{jl}, l = 1, 2, \dots, n$.

Кластеры G_k при $n = 2$ для $k = 2, 3, 4$ представлены соответственно на рис. 1, а, б, в соответственно.

Теорема 3.

Произведение кластеров $\{G_{k_i}(n_i, k_i)/i = 1, 2, \dots, s\}$ является кластером вида $G_k((n_1, n_2, \dots, n_s), (k_1, k_2, \dots, k_s))$ (рис. 2):

$$\prod_{i=1}^s G_{k_i}(n_i, k_i) = G_k((n_i/i = 1, 2, \dots, s), (k_i/i = 1, 2, \dots, s)). \quad (8)$$

Исследуем свойства кластеров в зависимости от значности логик k_i и размерности пространств n_i .

Нетрудно доказать следующие утверждения.

Теорема 4. Мощност носителя $|V(G_k)|$ кластера $G_k(n, k)$ равна k^n :

$$|V(G_k)| = k^n. \quad (9)$$

Следствие 1.

Мощност носителя кластера, равного декартову произведению кластеров $\{G_{k_i}(n_i, k_i)/i = 1, 2, \dots, s\}$, равна $\prod_{i=1}^s k_i^{n_i}$:

$$\left| V\left(\prod_{i=1}^s G_{k_i}(n_i, k_i)\right) \right| = \prod_{i=1}^s k_i^{n_i}. \quad (10)$$

Теорема 5. Кластер $G_k(n, k)$ является единственным в пространстве $P(n, K)$ и однородным степени $s(G_k)$, равной $(k - 1)^n$:

$$s(G_k) = (k - 1)^n. \quad (11)$$

Следствие 2. Запрещенной фигурой функциональной несвязности нетерминальных символов в пространстве $P(n, K)$ является звезда Q_3 степени $(k - 1)^n + 1$:

$$s(Q_3) = (k - 1)^n + 1. \quad (12)$$

Следствие 3. Для булевой логики, если граф сцепления $G_{сц}$ является паросочетанием (реберно-пустым графом), то существует функционально несвязное кодирование нетерминальных символов.

Теорема 6.

Кластер G_k , представляющий собой декартово произведение кластеров $\{G_{k_i}(n_i, k_i)/i = 1, 2, \dots, s\}$, является однородным степени $s(G_k)$, равной

$$\prod_{i=1}^s (k_i - 1)^{n_i}; \quad (13)$$

$$s(G_k) = \prod_{i=1}^s (k_i - 1)^{n_i}.$$

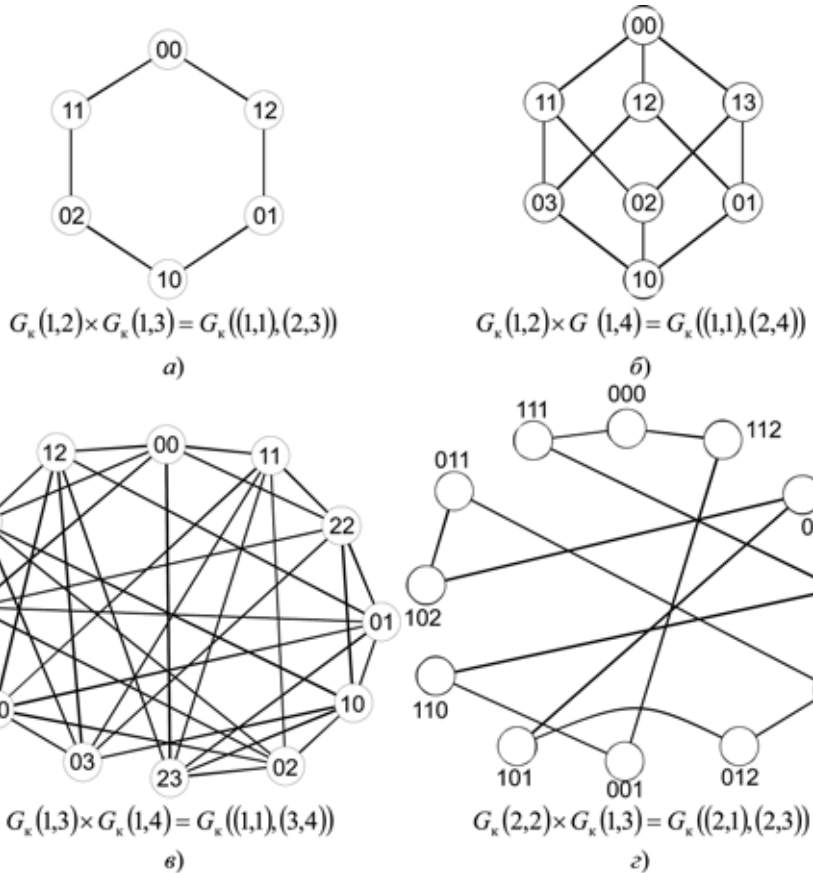


Рис. 2. Вид кластеров, получаемых в результате произведения кластеров

Теорема 7. Плотность $p(G_k(n, K))$ кластера $G_k(n, K)$ равна значности логики

$$p(G_k(n, K)) = k. \quad (14)$$

Доказательство. Значение каждого разряда z_l кодирующего вектора $Z(s_i)$, $z_l \in Z(s_i)$ получается путем перестановки, быть может циклической, значений k -значной логики $\{0, 1, 2, \dots, k - 1\}$. Следовательно, плотность кластера равна числу переставляемых значений, т. е. k , $p(G_k) = k$.

Следствие 4. Плотность $p(G_k)$ кластера, являющегося декартовым произведением кластеров $\{G_{k_i}(n_i, k_i)/i = 1, 2, \dots, s\}$, равна плотности кластера-сомножителя, у которого значность логики минимальна:

$$p(G_k) = \min_i k_i. \quad (15)$$

Коэффициентом сцепления $K_{сц}(G_k(n, K))$ кластера $G_k(n, K)$ называется максимальное число вершин, общих для двух полных подграфов $F_a, F_b \in G_k(n, K)$ плотности k :

$$K_{сц}(G_k(n, K)) = k - 2. \quad (16)$$

Найденные характеристика и структура кластеров позволяют эффективно определять суженные сигнатуры графа сцепления при минимизации

ции функциональной связности нетерминальных символов автоматной грамматики.

В результате, задача свелась к вложению максимального по числу ребер подграфа $\tilde{G}_{\text{сц}}$ графа сцепления $G_{\text{сц}}$ в кластер, однозначно определяемый размерностью пространства и значностью логики.

Вложение одного графа G_{α} в другой G_{β} сводится к определению максимальных по числу ребер изоморфных подграфов \tilde{G}_{α} , $\tilde{G}_{\alpha} \subset G_{\alpha}$ и \tilde{G}_{β} , $\tilde{G}_{\beta} \subset G_{\beta}$.

Список литературы

1. Горбатов В. А. Фундаментальные основы дискретной математики. Информационная математика. М.: Наука. Физматлит, 1999. 544 с.
2. Горбатов А. В. Характеризационная теория синтеза функциональных декомпозиций в k -значных логиках. М.: Изд-во физ.-мат. литературы, ООО "АСТ", 2000, 336 с.
3. Калитин Д. В. Автоматизированное проектирование минимальносвязной параллельной декомпозиции управляющих автоматов в k -значных логиках на основе теоретико-графового вложения. Автореферат диссертации на соискание ученой степени кандидата технических наук. МГГУ, 2005. 20 с.

ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.414.2

Т. И. Михеева, д-р техн. наук, проф.,
e-mail: mikheevati@mail.ru,
О. Н. Сапрыкин, аспирант,
e-mail: saprykinon@mail.ru,
Самарский государственный
аэрокосмический университет

Паттерны проектирования подсистемы анализа состояния пространственно-координированных объектов интеллектуальной транспортной системы

Определены требования, которым должна удовлетворять автоматизированная подсистема интеллектуального анализа состояния пространственно-координированных объектов интеллектуальной транспортной системы. Использование паттернов проектирования выбрано, как способ удовлетворения требованиям. Рассмотрены паттерны проектирования разных уровней и показан способ их применения в интеллектуальной транспортной системе.

Ключевые слова: геоинформационная система, класс, объект, клиент, сервер, сервис-ориентированное проектирование

Введение

Осознание и освоение парадигмы проектирования, в частности, применительно к транспортным системам, происходило практически параллельно с развитием электронной вычислительной техники. Принципы декомпозиции, абстракции и иерархии являются основополагающими в процессе проектирования, вследствие чего широкое распространение во всех инженерных дисциплинах получили объектно-ориентированный анализ,

проектирование и программирование. Методология формализации и анализа проблем интеллектуальной транспортной системы (ИТС) базируется на комплексной стратегии значительного повышения уровня абстракции используемых моделей, охватывающих реализуемые комбинации особенностей объектов предметной области. Кроме того, необходим учет фундаментальной базы знаний предметной области решаемой задачи в целях получения максимально содержательного конкретного результата этого анализа [1]. Такой методологии соответствуют концепции объектно-ориентированного анализа и проектирования, в основе которого лежит представление о том, что систему надо проектировать как совокупность взаимодействующих друг с другом объектов, рассматривая объекты как экземпляры определенных классов, образующих иерархию [2].

Термин "интеллектуальные транспортные системы" характеризует комплекс интегрированных средств управления транспортной инфраструктурой (улично-дорожной сетью, техническими средствами организации дорожного движения, транспортными потоками), применяемых для решения задач организации дорожного движения, на основе современных информационных технологий, организации информационных потоков о функционировании транспортной инфраструктуры в реальном режиме времени. Многоуровневая, сложноорганизованная ИТС представляет собой гибридную систему, состоящую из множества разнородных систем, сложным образом взаимодействующих друг с другом — управляющих, классифицирующих, прогнозирующих, экспертных, принимающих решения или поддерживающих эти процессы, объединенных для достижения единой цели [1].

Существенными особенностями проектируемой интеллектуальной транспортной системы, как информационной модели, являются:

- сложность и масштабность моделей, наполняющих ИТС, выражающиеся в большом числе типов, в применении альтернативных механизмов множественного наследования и полиморфного переопределения свойств объектных типов, в использовании вложенных агрегатных и селективных конструкций и двунаправленных ассоциаций;

- необходимость поддержки запросов к данным в декларативном, предикативном и навигационном стилях, эффективной реализации базовых операций манипулирования ими;
- широкий контекст использования моделей в приложениях, оперирующих как с данными одной многопрофильной информационной схемы, так и с данными нескольких независимых схем.

Формализованное описание задачи проектирования, ее решения, рекомендации по применению этого решения в различных ситуациях, моделируемых проектируемой системой, резонно реализовать паттерном проектирования. Это требование продиктовано тем, что проектируемая система должна удовлетворять новейшим требованиям к программному обеспечению, быть стабильной, масштабируемой и интероперабельной. Система должна быть рассчитана на дальнейшее изменение и развитие наряду с появлением новых алгоритмов и технологий. В связи с чем, необходимо избежать или, по крайней мере, свести к минимуму необходимость перепроектирования. Все это можно обеспечить, применив паттерны проектирования программных систем [3].

1. Геоинформационная составляющая ИТС

ИТС работает с большим количеством разнородной информации, поэтому одним из требований, предъявляемых к системе, является скорость обработки данных. В целях применения для обработки информации наиболее мощных средств в ИТС принято решение о разделении хранилища данных на две части: геоинформационной и семантической составляющие базы данных (БД). Данными управляют соответствующие подсистемы, мастер доступа к данным реализует их консолидацию на уровне бизнес-логики и декомпозицию на уровне БД.

Геоинформационные данные об объектах ИТС (вид геообъекта, координаты, площадь, топология и т. д.) хранятся в базе данных под управлением геоинформационной СУБД. С помощью геоинформационной системы (ГИС) организуется эффективный доступ к большому объему информации об объектах, имеющих пространственную привязку. Скорость отображения и удобство анализа информации с помощью ГИС достигается благодаря возможности выборочного, послойного отображения информации. Почти все современные векторные ГИС используют методологию "слоев" данных, объединяя объекты одного типа в отдельные таблицы — слои и устанавливая связи между ними. Такие геореляционные системы отображают только нужную часть слоев в один момент времени и определенный набор объектов внутри каждого слоя, удовлетворяющих заданным критериям.

Географическая информация, хранящаяся в системе, состоит из объектов двух типов: объектов электронной карты и объектов модели ИТС. Объекты модели ИТС являются продуктом функционирования системы и одновременно источником для получения новых данных. Электронная карта представляет собой первичную исходную информацию о регионе, отображаемую в информационных слоях и представленную текстом (названия улиц, районов и т. д.) или различными графическими примитивами: полигоном (районы, кварталы, строения, проезжие части дорог); полилинией (осевые линии улиц, транспортные маршруты, железные дороги); эл-

липсом (остановки общественного транспорта). Помимо общезначимых геоинформационных данных электронная карта содержит данные об объектах ИТС: полигоны (перекрестки, перегоны, пешеходные переходы, ж/д переезды); линии (дуги орграфа транспортной сети, дорожная разметка); точки (координаты дислокации дорожных объектов: светофорных объектов, дорожных знаков, световых опор и т. п.); поверхности (очаги аварийности, уровень интенсивности движения транспортных потоков).

Семантическая информация хранится в базе данных под управлением серверной СУБД. Это делает возможным применения эффективных алгоритмов поиска и обработки данных, высокоскоростного обмена данными, встроенных в СУБД средств резервного копирования информации и разграничения прав пользователей.

В последние годы развитие ГИС идет одновременно в нескольких направлениях. Во-первых, ГИС стали более доступны за счет сервисов популярных поисковых систем (например, Google и Яндекс). Во-вторых, увеличивается точность дистанционного зондирования с помощью искусственных спутников Земли (GeoEye, WorldView и FormoSat). В-третьих, профессиональные геоинформационные системы начали оснащать средствами пространственного анализа, что расширяет круг пользователей и областей применения.

ИТС в качестве подсистемы содержит автоматизированную аналитическую систему обработки и анализа пространственно-координированных объектов ИТС. Система имеет комплекс аналитических инструментов, позволяющих решать нетривиальные задачи на электронных картах: кластерный анализ, регрессионный анализ, классификация и поиск ассоциативных правил. При решении задач используются как статистические алгоритмы, так и алгоритмы искусственного интеллекта. Использование тех или иных методов должно обеспечить их корректную работу с пространственными данными [4, 5].

2. Паттерны проектирования программных систем

При проектировании многоуровневой системы требуется выявить объекты предметной области, отнести их к классам, определить интерфейсы классов и иерархию наследования, установить регламент отношений между классами, выделить компоненты системы, распределить классы между компонентами, определить контракты взаимодействия компонентов внутри системы и самой системы с внешней средой. Архитектура должна отражать потребности решаемой задачи, но, в то же время, быть общей, чтобы соответствовать требованиям к системе, которые могут возникнуть в будущем. Во многих современных системах можно встретить шаблоны, состоящие из классов, взаимодействующих объектов, а также содержащих их компонентов и сервисов, с помощью которых можно решать конкретные задачи проектирования, распространенные во многих системах. Обобщение и классификация таких задач и наиболее удачных путей их решения привели к появлению паттернов [6].

Паттерн проектирования — шаблонная модель, формализованное описание часто встречающейся задачи проектирования, эффективное, в определенном контексте, типовое решение проектной (программной) пробле-

мы, а также рекомендации по применению этого решения в различных ситуациях [7].

Каждый паттерн описывает некую повторяющуюся проблему и метод ее решения, причем таким образом, что этим методом можно пользоваться при решении самых разнообразных задач. Паттерн проектирования именуется, абстрагирует и идентифицирует ключевые аспекты структуры общего решения, которые и позволяют применить его для создания повторно используемой архитектуры. Данная особенность паттернов позволяет при проектировании ИТС абстрагироваться от особенностей реализации и сконцентрироваться на сущностях предметной области и решаемой задаче.

Использование паттернов проектирования дает неоспоримые преимущества: построенная в терминах паттернов проектирования модель системы является структурированным выделением тех элементов и связей, которые значимы при решении поставленной задачи. Модель позволяет глубоко и всесторонне проработать архитектуру разрабатываемой системы с использованием специального языка. Применение паттернов проектирования повышает устойчивость системы к изменению требований и упрощает последующую доработку системы.

По уровню использования паттерны можно разделить на следующие группы:

- паттерны проектирования классов и объектов;
- архитектурные системные паттерны;
- паттерны интеграции информационных систем [7].

При проектировании ИТС учтены все уровни использования паттернов проектирования и особенности каждого из них рассмотрены на примере архитектуры подсистемы ИТС, предназначенной для анализа пространственно-координированных объектов в ГИС.

3. Реализация паттернов проектирования в архитектуре автоматизированной системы анализа пространственно-координированных объектов

Для удовлетворения перечисленных требований к автоматизированной аналитической системе необходимо корректно подобрать паттерны проектирования и их взаимные связи на всех уровнях использования. Описание архитектуры в терминах паттернов проектирования позволяет абстрагироваться от технической реализации таких особенностей проекта как распределенное вычисление, разделение системы на уровни, адаптивное изменение данных для обработки на разных уровнях и другие.

3.1. Использование паттернов проектирования классов и объектов

Описание системы в терминах классов и объектов является низшим уровнем ее представления. По степени абстракции паттерны проектирования классов и объектов можно разделить на:

- паттерны распределения обязанностей;
- паттерны взаимодействия классов.

Паттерны *распределения обязанностей* имеют своей целью показать общие принципы проектирования объектно-ориентированных систем, не вдаваясь в возможные варианты использования. К таким паттернам можно отнести боль-

шинство паттернов *General Responsibility Assignment Software Patterns (GRASP)* [6].

Паттерны *взаимодействия классов* ориентированы на более частные случаи проектирования, под конкретные варианты использования. К ним можно отнести некоторые паттерны *GRASP* и все паттерны *Gang-of-Four (GoF)* [7].

В ходе проектирования автоматизированной системы анализа пространственно-координированных объектов ИТС использовались следующие паттерны: Адаптер (GoF), Заместитель (GoF), Информационный эксперт (GRASP), Низкая связность (GRASP), Итератор (GoF), Наблюдатель (GoF), Стратегия (GoF), Высокое зацепление (GRASP), Контроллер (GRASP), Полиморфизм (GRASP), Одиночка (GoF), Создатель (GRASP), Фабричный метод (GoF). Рассмотрим некоторые из них более подробно в применении к задачам, решаемым в рамках ИТС.

Создатель. В силу того что аналитическая система должна быть масштабируемой, она должна соответствовать архитектуре "клиент-сервер". В архитектуре "клиент-сервер" клиент не обладает достаточными знаниями о том, как создавать объекты на сервере. Клиент имеет только набор интерфейсов серверных объектов и ссылку на заместителя серверного объекта-одиночки (паттерны *Одиночка* и *Заместитель* будут описаны далее). Наилучшим выходом в этой ситуации является применение паттерна *Создатель (Creator (GRASP))*. Основное назначение — выявление объекта-создателя, который связан со всеми созданными им объектами. В качестве объекта-создателя в данном случае будет выступать объект-одиночка. *Создатель* инкапсулирует методы создания объектов алгоритмов и моделей, а также вспомогательных объектов. На рис. 1 изображена диаграмма классов, показывающая использование паттерна проектирования *Создатель* в ИТС.

Создателем в данном случае является класс *АнализаторДанных*, который реализует интерфейс *IAнализаторДанных*. Как видно из рис. 1, клиентский элемент управления *УровеньАварийностиЭлУпр* ссылается только на интерфейсы серверных классов. Клиент, используя ин-

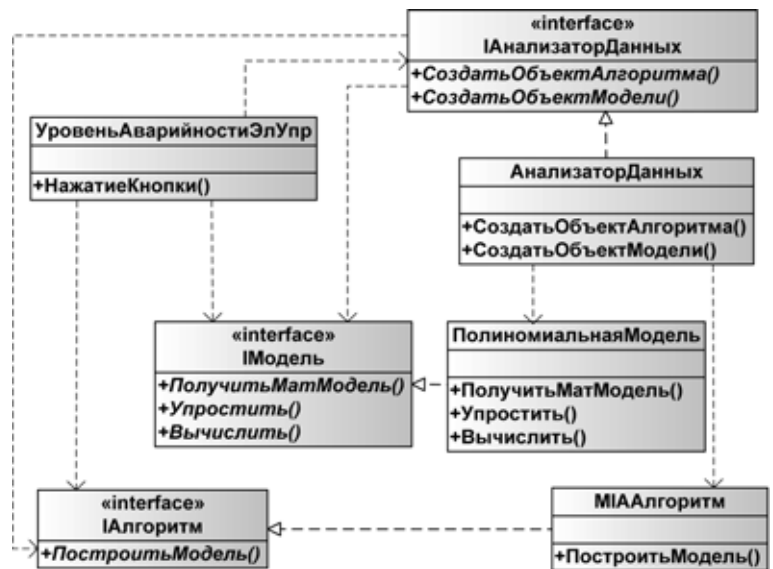


Рис. 1. Применение паттерна проектирования *Создатель*

терфейс **IAнализаторДанных**, создает на сервере единственный экземпляр класса **АнализаторДанных**, и последующее создание объектов возлагается на него.

Паттерн *Создатель* обеспечивает низкую степень связности.

Одиночка. Архитектура "клиент-сервер" подразумевает наличие на сервере объекта, синхронизирующего действия подключенных к нему клиентов. Для того чтобы гарантировать целостность системы во время ее функционирования, необходима уверенность в том, что синхронизирующий объект всегда будет только в одном экземпляре. Применение паттерна проектирования *Одиночка* (*Singleton (GoF)*) способно решить данную задачу.

Паттерн *Одиночка* гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа. Сам класс контролирует то, что у него есть только один экземпляр, может запретить создание дополнительных экземпляров, перехватывая запросы на создание новых объектов, и он же способен предоставить доступ к своему экземпляру.

От класса *Одиночки* можно порождать подклассы, а приложение легко сконфигурировать экземпляром расширенного класса. Можно конкретизировать приложение экземпляром того класса, который необходим во время выполнения.

Заместитель. Для корректного распределенного функционирования системы необходимо, чтобы клиент мог удаленно вызывать методы серверного класса. При этом необходимо для клиента обеспечить прозрачность подобных вызовов, т. е. вызов методов удаленного объекта не должен отличаться от вызовов методов локальных объектов. Данную задачу можно решить, применив паттерн проектирования *Заместитель* (*Proxy (GoF)*).

Объект-заместитель является суррогатом другого объекта и контролирует доступ к нему. Заместитель серверного объекта на клиентской стороне подставляется вместо реального серверного объекта, расположенного на сервере. *Заместитель* ведет себя так же, как сам серверный объект, и перенаправляет все вызовы на сервер. На рис. 2 изображена диаграмма классов, которая показывает применение паттерна *Заместитель*.

Класс **ЗаместительАнализатораДанных** является заместителем серверного объекта **АнализаторДанных**. Оба класса реализуют один интерфейс **IAнализаторДанных**, поэтому для клиента не видна разница между работой с реальным классом и его заместителем. Экземпляр класса **ЗаместительАнализатораДанных** находится на клиентской стороне и перенаправляет вызовы по сети на сервер.

Адаптер. Картографическая информация представлена на электронной карте в виде слоев (в *MapInfo* это таблицы в формате *.tab), на каждом из которых хранится определенный вид пространственных объектов. Алгоритмы анализа коррелируемых пространственных данных (пространственно-координированных) разработаны как универсальные и требуют в качестве входных параметров векторы и матрицы определенной размерности. Паттерн проектирования *Адаптер* (*Adapter (GoF)*) используется для преобразования информации из картографической формы в форму, удобную для работы нейроалгоритмов выявления зависимостей в пространственно-координированных данных. Паттерн преобразует интерфейс одного класса в интерфейс другого — того, который ожидают клиенты. Адаптер обеспечивает совместную работу классов с несовместимыми интерфейсами, которая без него была бы невозможна [7].

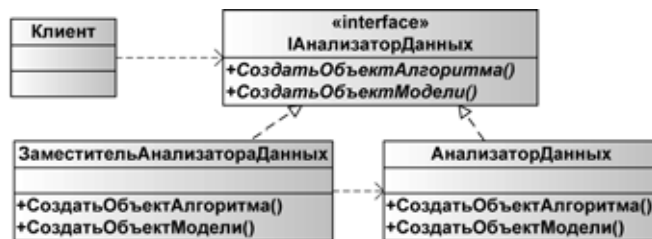


Рис. 2. Применение паттерна проектирования *Заместитель*

Вместо использования в методах картографической информации (что нарушает универсальность алгоритмов исследования данных), используется класс-адаптер. Адаптер агрегирует пространственно-координированные данные в их естественной форме (в виде слоев, таблиц) и оперирует с методами, возвращающими матрицы и векторы. Диаграмма классов для системы интеллектуального анализа состояния пространственно-координированных объектов показана на рис. 3.

Алгоритм — интерфейс, который наследуют все алгоритмы исследования данных. **АдаптерКарты** — адаптер, приводящий интерфейс пространственно-координированных объектов, хранящихся в экземплярах классов **Таблица** и **СлойКарты**, к интерфейсу **Алгоритм**. **Таблица**, **СлойКарты** — классы, описывающие сущности таблиц и слоев электронных карт — классы объектной модели *MapXtreme*.

Стратегия. При проектировании автоматизированной подсистемы интеллектуального анализа состояния пространственно-координированных объектов ИТС не-



Рис. 3. Применение паттерна проектирования *Адаптер*



Рис. 4. Применение паттерна проектирования *Стратегия*

обходимо учесть расширяемость системы в плане добавления новых алгоритмов исследования данных. Использование новых алгоритмов не должно отличаться от использования уже существующих. Система интеллектуального анализа состояния геообъектов позволяет определить зависимость уровня аварийности участка улично-дорожной сети от различных факторов, постоянно присутствующих или возникающих в процессе функционирования транспортной инфраструктуры: наличие детских учебных заведений, пешеходных переходов, дорожных знаков, аварийно-восстановительных работ, ведущихся на проезжей части, дорожно-транспортных происшествий и др.

Применение паттерна проектирования *Стратегия (Strategy (GoF))* позволило элегантно решить эту задачу. Паттерн определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми. *Стратегия* позволяет изменять алгоритмы независимо от клиентов, которые ими пользуются. На рис. 4 изображена диаграмма классов, которая показывает применение паттерна *Стратегия* в ИТС.

Клиентский класс элемента управления **Уровень Аварийности ЭлУпр** (обеспечивает определение уровня аварийности участка улично-дорожной сети) имеет доступ к интерфейсной сборке и поэтому может определить перемещенный тип **Алгоритм**. Каждый алгоритм описан в отдельном классе (**МИААлгоритм**, **СОМБИАлгоритм**, **ТМННАлгоритм**), реализуемым интерфейсом **Алгоритм**. Клиентский класс получает ссылку на один из алгоритмов, расположенных на сервере, посредством класса-создателя, описанного ранее. Решение о том, какой класс алгоритма создавать, выносит пользователь, вызывая один из перегруженных методов класса-создателя.

В системе реализованы алгоритмы метода группового учета аргументов:

- итерационный многоярдный алгоритм (*Multilayered iterative algorithm*, МИА) в классе **МИААлгоритм**;
- комбинаторный алгоритм (*Combinatorial algorithm*, СОМБИ) в классе **СОМБИАлгоритм**;
- дважды многоярдные нейронные сети (*Twice-Multilayered Neural Nets*, ТМНН) в классе **ТМННАлгоритм**.

С помощью паттерна *Стратегия* обеспечивается расширяемость системы за счет добавления новых алгоритмов построения регрессионной модели, добавление нового класса алгоритма к уже имеющимся не нарушит общей архитектуры системы.

3.2. Использование архитектурных системных паттернов

Архитектурные системные паттерны определяют эффективные пути объединения классов и объектов в компоненты и подсистемы, а также типовые способы взаимодействия между последними. В ходе проектирования использовались паттерны: *Репозиторий*, *Клиент-сервер*, *Модель предметной области*, *Многоуровневая система*, *Передача сообщений*, *Преобразователь данных*. Рассмотрим некоторые из них более подробно в применении к ИТС.

Клиент-сервер. На этапе проектирования необходимо предусмотреть возможность работы нескольких операторов (аналитиков) с подсистемой интеллектуального анализа состояния пространственно-координированных объектов ИТС. Они должны работать с одним набором данных и их действия должны быть синхронизированы.



Рис. 5. Трехуровневая архитектура

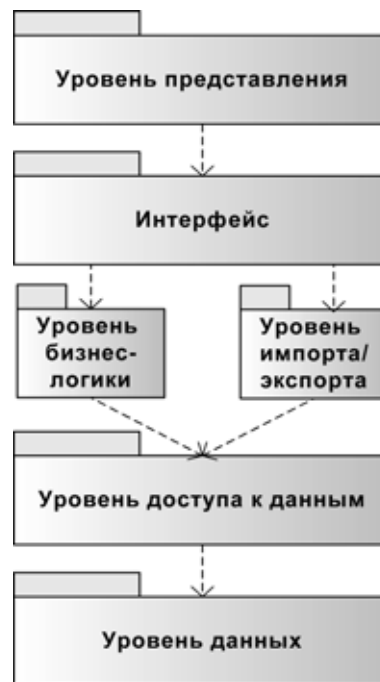


Рис. 6. Многоуровневая система

Эти задачи в ИТС решаются с помощью архитектуры "клиент-сервер". Архитектура постоянно эволюционирует, наиболее устойчивым ее решением является трехуровневая архитектура (рис. 5).

Клиентские приложения ответственны за уровень представления (ввод пользователем данных и визуализация результата). База данных ответственна за хранение данных и обеспечение доступа к ним. Сервер приложений состоит из уровней бизнес-логики и данных. Уровень бизнес-логики верифицирует данные, пришедшие от клиента, и обрабатывает их в соответствии с бизнес-правилами. Обработка может содержать взаимодействие с уровнем данных и выполнение локальных вычислений. Промежуточный уровень передает результаты на уровень данных для хранения или возвращает клиенту результаты. Преимущество такой архитектуры заключается в более четком по сравнению с другими архитектурами распределении ответственности при обработке данных.

Если более одного уровня многоуровневой системы находится на одном компьютере, то разделение функций системы также дает преимущества. Разработчики получают возможность работать с разными уровнями по отдельности, например, переносить их на другие ком-

пьютеры в соответствии с новыми требованиями масштабируемости [8].

Многоуровневая система. Паттерн предназначен для разграничения ответственности между элементами системы. На рис. 6 изображены уровни подсистемы интеллектуального анализа состояния пространственно-координированных объектов ИТС. Уровень данных содержит электронную карту и базу данных с атрибутивной информацией. Электронная карта содержит набор файлов, описывающих пространственно-координированные данные, и предоставляет к ним программный интерфейс в виде *OLE DB* провайдеров.

Сервер приложений обеспечивает:

- загрузку электронной карты в оперативную память;
- работу алгоритмов анализа;
- синхронную работу клиентов.

Используя уровень доступа к данным на сервере приложений, можно повысить масштабируемость путем увеличения числа одновременных подключений к БД. Уровень данных может поддерживать пул соединений с базой данных и предоставлять их клиентам при необходимости непосредственного доступа к БД. По завершении работы с БД соединение не закрывается, а возвращается в пул соединений. Таким образом, небольшое число постоянно поддерживаемых соединений в состоянии обслуживать большое число клиентов. Интерфейс содержит типы (интерфейсы и классы), доступные как серверу, так и клиенту. По этой причине интерфейс дублируется на сервере и на клиентских станциях.

Клиентское приложение (уровень представления) обеспечивает пользовательский интерфейс, позволяющий пользователю управлять процессами на сервере. Система имеет несколько уровней представления, которые различаются как по функциональности, так и средой выполнения. Например, существуют администраторские инструменты, редакторы карт и просмотрщики. Клиентские приложения могут быть разработаны под разные операционные системы (*Microsoft Windows, Mac OS X, Linux*), выполняться в Интернет-браузере или на карманных персональных компьютерах.

3.3. Использование паттернов интеграции информационных систем

В последнее время наблюдается рост интеграции ГИС в глобальное информационное пространство Интернет. Повышения эффективности данного процесса можно достигнуть путем взаимодействия с сервисами в глобальной сети. Например, сервисами полезными для подсистемы интеллектуального анализа состояния пространственно-координированных объектов ИТС являются сервисы электронных карт и сервисы спутниковых снимков. Между тем, организовав собственную систему как веб-сервис, можно предоставить к ней доступ третьих систем (рис. 7).

Структуру и механизмы межсистемного взаимодействия определяют паттерны интеграции информационных систем. Данные паттерны находятся на верхней ступени в иерархии паттернов проектирования.

При проектировании применяется сервис-ориентированный подход. Сервис-ориентированное проектирование — это архитектурный подход, при котором интеграция слабосвязанных компонентов распределенных

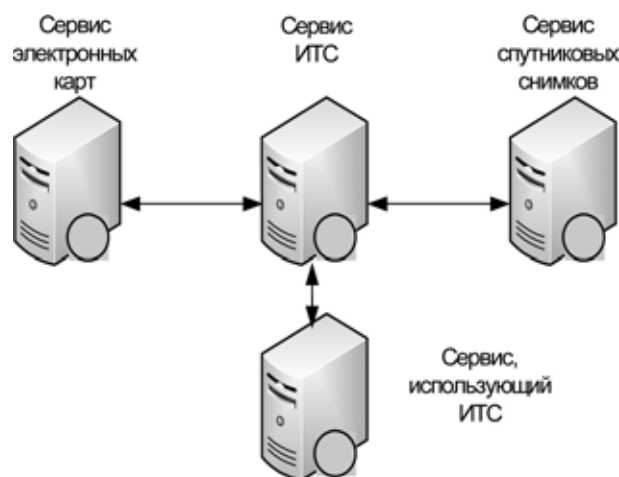


Рис. 7. Интеграция информационных систем

приложений обеспечивается посредством сообщений и контрактов.

В архитектуре разрабатываемой системы применяется паттерн проектирования *Обмен Сообщениями*. В сервис-ориентированных приложениях фундаментальной единицей взаимодействия являются сообщения. Для реализации паттерна необходимо распределить между участниками процесса следующие роли: исходный отправитель сообщений, посредник и конечный приемник сообщений. Принимающей и отправляющей сторонам необходима следующая информация для установки контакта: адрес конечной точки, кодировка и контракт.

Совместимость осуществляется за счет поддержки системами таких стандартов как *HTTP, URL, XML* и *WS-**. Если первые три стандарта известны уже продолжительное время и поддерживаются подавляющим большинством систем, то последний разработан сравнительно недавно, и его поддержка системами может быть ограниченной или вообще отсутствовать. *WS-** — это семейство стандартов, описывающих формат сообщений, их передачу, шифрование, транзакционность пересылки и адресацию конечных точек. При отсутствии поддержки данного стандарта используется паттерн *Адаптер* для преобразования одного формата сообщений в другой.

С ростом числа пользователей может сложиться ситуация, когда система перестанет отвечать на запросы, потому что пространственные аналитические задачи очень трудоемкие и требуют больших вычислительных ресурсов. Решить проблему можно, используя парадигму **"облачных вычислений" (Cloud Computing)**, в которой информация хранится и обрабатывается на сетевых масштабируемых виртуальных ресурсах, доступ к которым осуществляется по запросу. Согласно этой парадигме все используемые ресурсы представляются в виде сервисов: *Аппаратное обеспечение (Hardware as a Service — HaaS), Программное обеспечение (Software as a Service — SaaS), Платформа (Platform as a Service — PaaS) и База данных (Data as a Service — DaaS)* [9].

Используя *Облачные вычисления*, можно решить проблему с помощью высокоуровневой платформы в виде сервиса. Эта платформа предоставляет возможность создавать собственные сервисы ИТС из составных блоков и размещать базы данных с географической информацией в виде сервиса (рис. 8).

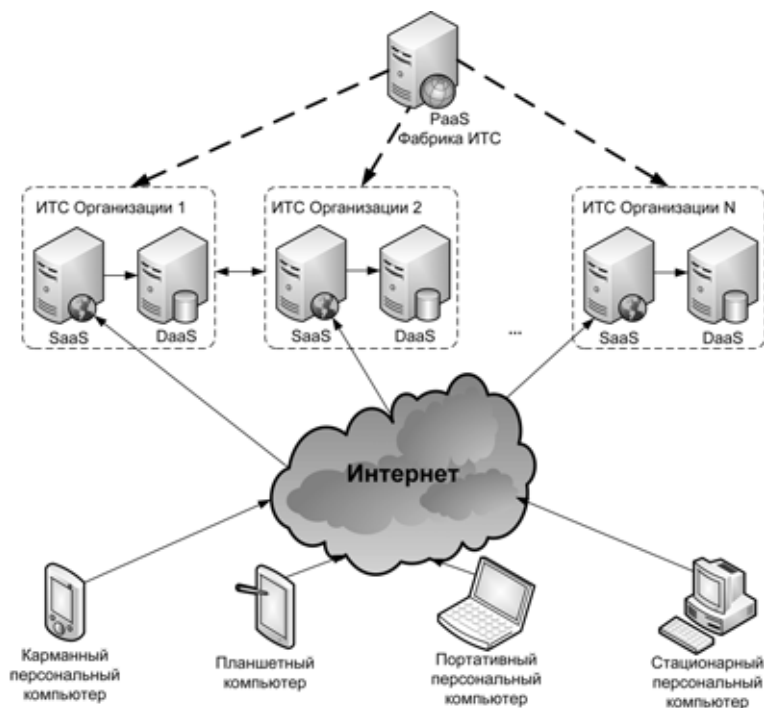


Рис. 8. Виртуализированные серверы ИТС

Сотрудник из заинтересованной организации, используя сервис *Фабрика ИТС*, создает собственный сервис ИТС из базового блока и дополнительных блоков (аналитических, экспорта/импорта, преобразования и т. д.). Он выделяет необходимую область электронной карты, и сервис *Фабрика ИТС* копирует ее как отдельный источник данных в отдельный сервис, наряду с этим создается база данных, имеющая заданную структуру для хранения проектов, информации о пользователях и атрибутивной информации электронной карты. Сотрудник имеет возможность изменить настраиваемые параметры системы и создать пользовательские учетные записи.

Сотрудники, получившие доступ к сервису, могут работать с ним на любом компьютере, имеющем доступ в Интернет. Это может быть офисный стационарный компьютер, портативный компьютер или карманный персональный компьютер, что дает свободу выбора сценария работы с ИТС. При этом организация освобождается от необходимости поддержки серверов. Вся ответственность за работоспособность сервиса и сохранность данных ложится на провайдера виртуальных ресурсов.

Все ИТС-серверы, созданные с помощью *Фабрики ИТС*, являются однотипными и, следовательно, поддерживают паттерн *Обмен Сообщениями*. Это позволяет им не только обмениваться информацией с произвольным числом сторонних сервисов, но также интегрировать не-

сколько ИТС-серверов. При этом объединение может быть на разном уровне: от простого обмена данными и моделями до объединения серверов в единую *grid*-инфраструктуру.

Заключение

Применение описанных в статье паттернов проектирования позволило создать стабильную, масштабируемую и интероперабельную подсистему интеллектуального анализа состояний пространственно-координированных объектов ИТС. Проект системы рассчитан на постоянное развитие системы, в ней упрощена процедура добавления новых алгоритмов, модулей и источников данных. Проект учитывает масштабирование системы от локальной установки до использования в глобальной сети Интернет. Разделение системы на уровни позволяет определить качество каждого из них по отдельности, быстро локализовать ошибку и обеспечивает стабильную работу системы. Интероперабельность системы обеспечивает использование общепризнанных стандартов интеграции приложений. В результате исключено перепроектирование системы в будущем, которое вносит существенные затраты на этапе выполнения проекта.

Список литературы

1. **Михеева Т. И.** Использование принципов объектно-ориентированного проектирования интеллектуальной транспортной системы // Вестник Самарского гос. техн. ун-та. Сер. Физико-математические науки. 2004. № 34. С. 141–149.
2. **Буч Г.** Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Пер. с англ. — СПб.: Невский диалект, 1999. 560 с.
3. **Сапрыкин О. Н.** Паттерны проектирования в архитектуре ИТС // Математика. Компьютер. Образование: Тезисы докладов XV международной конф. — М.-Ижевск: НИЦ "Регулярная и хаотическая динамика", 2008. С. 116.
4. **Михеева Т. Н., Сапрыкин О. Н.** Применение нейросетевых методов для анализа пространственных данных // Организация и безопасность дорожного движения в крупных городах: Труды Седьмой международной научно-практической конференции. СПб.: СПб. гос. архит.-строит. ун-т, 2006. С. 81–84.
5. **Михеева Т. И., Сапрыкин О. Н.** Идентификации зависимостей в пространственно-распределенных данных с использованием нейросетевых технологий // Вестник Самарского гос. техн. ун-та. Сер. Технические науки. 2007. № 1 (19). С. 40–47.
6. **Ларман К.** Применение UML и шаблонов проектирования: пер. с англ. М.: Вильямс, 2004. 624 с.
7. **Приемы** объектно-ориентированного проектирования. Паттерны проектирования / Гамма Э., Хелм Р., Джонсон Р. и др. — СПб.: Питер, 2001. 368 с.
8. **Оберг Дж.** Технология COM+. Основы и программирование: уч. пособие, пер. с англ.: М.: Вильямс, 2000. 480 с.
9. **Смит Дж.** Основы Windows Communication Foundation: пер. с англ. М.: Русская Редакция; СПб.: БХВ-Петербург, 2008. 384 с.

А. Н. Чичко, д-р физ.-мат. наук, проф.,
e-mail: chichko_a_n@mail.ru,

О. А. Сачек, аспирант,
e-mail: OSachok@mail.ru,

С. Г. Лихоузов, канд. техн. наук, ст. науч. сотр.,
e-mail: slh@tut.ru,

Белорусский национальный технический
университет, г. Минск

Методы автоматизации обработки изображений микроструктур перлитных сталей

Описаны математические методы для обработки изображений микроструктур перлитных сталей. Методы, позволяющие проводить количественный и качественный анализ характеристик микроструктуры на основе межпластиночных расстояний фаз, основаны на аппроксимации микроструктуры функциями плотности распределения межпластиночных расстояний. Показано, что микроструктуры перлитной стали могут быть разделены на два класса на основе функций распределения межпластиночных расстояний.

Ключевые слова: обработка изображений, микроструктуры, перлитная сталь, функция распределения межпластиночных расстояний, дисперсность перлита

Известно, что исследование микроструктур сплавов в целях изучения механических свойств является одной из важнейших задач физики металлов и металловедения [1]. Как правило, изображение микроструктуры формируется от фазовых составляющих сплава в отраженном свете, если микроскоп световой, и во вторичных электронах, если микроскоп электронный [2]. Полученные с помощью методов световой или электронной микроскопии изображения по форме представляют собой сложное плоскостное многопиксельное изображение, по которому можно оценивать размеры зерен и фазовых составляющих [3]. В настоящей статье развиваются математические методы количественного анализа изображений микроструктур перлитных сталей, используемых при изготовлении проволок [4].

Известно, что микроструктуры перлитных сталей, полученных в неравновесных условиях, состоят из эвтектоидных колоний, образовавшихся после распада аустенита и неоднородно распределенных в плоскости шлифа. При этом в различных участках микроструктуры размеры межпластиночного расстояния системы "феррит — цементит" могут различаться в несколько раз. В этих

условиях установление взаимосвязей между характеристиками микроструктуры и свойствами стали практически невозможно, если использовать традиционные подходы, основанные на качественном анализе микроструктуры. Простое усреднение параметров микроструктуры сглаживает особенности строения эвтектоидных колоний, и исследование взаимосвязей между свойствами стали и его параметрами микроструктуры еще более усложняется. В то же время современные компьютерные методы обработки изображений открывают новые возможности для исследования взаимосвязей "структура — свойства".

Цель настоящей статьи — разработка метода формализации и обработки изображений микроструктур перлитных сталей для их системного анализа и классификации в целях выявления взаимосвязей "свойства — параметры микроструктуры".

Известно, что фотографическое изображение микроструктуры содержит в себе гораздо больше информации, чем та, которую традиционно используют металловеды в своих научных исследованиях. Как правило, это качественная информация о форме фазовых скоплений либо усредненная информация о размере зерен или площадях фаз. При этом следует отметить, что даже в рамках одной микроструктуры форма фазовых скоплений и размеры зерен варьируются в широких пределах, особенно для неравновесно охлажденных сплавов. Основной научной идеей, развиваемой в статье, является разработка метода обработки изображения микроструктуры на основе вычислений статистической функции плотности распределения межпластиночных расстояний, на базе которой рассчитываются характеристики микроструктуры, используемые для ее оценки.

Сущность предлагаемого метода состоит в том, что изображение микроструктуры заменяется множеством характеристик функции распределения, которая отражает особенности строения фазовых составляющих микроструктуры.

Методы исследований

Для построения функции плотности распределения микроструктуры предлагается следующий математический аппарат.

Первоначально изображение представляется в виде множества пикселей p_{ij} , имеющих различные значения уровней цвета, вычисленных по формуле

$$N_{ij}^0 = r \cdot 255^2 + g \cdot 255 + b,$$

где $r, g, b \in [0; 255]$ — уровни соответственно красного, зеленого и синего в палитре RGB ; i, j — ин-

дексы двумерного пространства в интервалах $1 \leq i \leq L_x, 1 \leq j \leq L_y$; L_x, L_y — размеры изображения в пикселях по осям координат X и Y соответственно.

На первом этапе из исходного изображения микроструктуры (множество $\Omega^0 = \{N_{ij}^0 \in N | 0 \leq N_{ij}^0 \leq 16646655\}$) с помощью функций пороговой обработки с использованием формулы

$$N_{ij}^1 = \frac{N_{ij}^0}{\max\{N_{ij}^0\}} \quad (1)$$

создается полутоновое изображение микроструктуры (множество $\Omega^1 = \{N_{ij}^1 \in R | 0 \leq N_{ij}^1 \leq 1\}$).

Далее с помощью преобразования (2) изображение трансформируется в вектор длиной m бинаризованных изображений $\Omega^2 = \{\Omega_1^2, \Omega_2^2, \dots, \Omega_m^2\}$, где $\Omega_k^2 = \{N_{ijk}^2 | N_{ijk}^2 = -1 \text{ или } N_{ijk}^2 = 1\}$:

$$N_{ijk}^2 = \begin{cases} -1, & \text{если } N_{ij}^1 \leq \beta_k; \\ 1, & \text{если } N_{ij}^1 > \beta_k. \end{cases} \quad (2)$$

Здесь $\beta_k \in \{\beta_1, \beta_2, \dots, \beta_m\}$ — порог бинаризации; k — индекс вектора параметров α , полученных с помощью формулы (3), для которого выполняется условие (4):

$$\alpha_k = \frac{a_k^{-1}}{a_k^1}; \quad (3)$$

$$a_k^{-1} = \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} t_{ijk}, \text{ где } \begin{cases} t_{ijk} = 1, & \text{если } N_{ij}^3 = -1; \\ t_{ijk} = 0, & \text{если } N_{ij}^3 = 1; \end{cases}$$

$$a_k^1 = \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} t_{ijk}, \text{ где } \begin{cases} t_{ijk} = 1, & \text{если } N_{ij}^3 = 1; \\ t_{ijk} = 0, & \text{если } N_{ij}^3 = -1; \end{cases}$$

$$\alpha_k - \frac{\varepsilon}{2} < \alpha^p < \alpha_k + \frac{\varepsilon}{2}. \quad (4)$$

Здесь α^p — заданное значение; ε — отклонение α^p .

Полученный вектор изображений с использованием формулы

$$N_{ij}^3 = \begin{cases} 0, & \text{если } N_{ijk}^2 = -1; \\ 1, & \text{если } N_{ijk}^2 = 1 \end{cases} \quad (5)$$

преобразуется в модифицированное бинаризованное изображение микроструктуры $\Omega^3 = \{N_{ij}^3 | N_{ij}^3 = 0 \text{ или } N_{ij}^3 = 1\}$.

Затем проводится сканирование изображения в целях определения межпластиночных расстояний микроструктуры, которое состоит из нескольких этапов:

1. Поиск базового пикселя p_{ij} , удовлетворяющего условию $N_{ij}^3 = 1$.

2. Вычисление межпластиночного расстояния $d_{ij}^{\text{пл}}$, содержащего базовый пиксель p_{ij} , по формуле

$$d_{ij}^{\text{пл}} = \frac{(d_1 + d_2)(d_3 + d_4)}{\sqrt{(d_1 + d_2)^2 + (d_3 + d_4)^2}},$$

где d_1 — значение, удовлетворяющее условию

$$\begin{cases} N_{ij+d_1+1} = 0; \\ N_{ij+w} = 1, w = \overline{1, d_1}; \end{cases} \quad (6)$$

d_2 — значение, удовлетворяющее условию

$$\begin{cases} N_{(i+d_2+1)j} = 0; \\ N_{(i+w)j} = 1, w = \overline{1, d_2}; \end{cases} \quad (7)$$

d_3 — значение, удовлетворяющее условию

$$\begin{cases} N_{ij-d_3-1} = 1; \\ N_{ij-v} = 1, v = \overline{1, a}; \\ N_{ij-w} = 0, w = \overline{a+1, d_3}; \end{cases} \quad (8)$$

d_4 — значение, удовлетворяющее условию

$$\begin{cases} NN_{(i-d_4-1)j} = 1; \\ N_{(i-v)j} = 0, w = \overline{1, a}; \\ N_{(i-w)j} = 0, w = \overline{a+1, d_4}. \end{cases} \quad (9)$$

Таким образом, вычисление видимого межпластиночного расстояния $d_{ij}^{\text{пл}}$ базируется на определении четырех расстояний:

- расстояния d_1 вниз от базового пикселя p_{ij} до первого пикселя феррита при условии, что до него все пиксели принадлежат одной цементитной пластине — формула (6);
- расстояния d_2 вправо от базового пикселя до первого пикселя феррита при условии, что до него все пиксели принадлежат одной цементитной пластине — формула (7);
- расстояния d_3 вверх от базового пикселя до первого пикселя следующей цементитной пластины при условии, что до нее имеется пластина феррита — формула (8);
- расстояния d_4 влево от базового пикселя до первого пикселя следующей цементитной пластины при условии, что до нее имеется пластина феррита — формула (9).

В результате будет получено множество расстояний для всех базовых пикселей p_{ij} изображения $\Omega^4 = \{d_{ij}^{\text{пл}} \in R\}$.

Затем составляется функция распределения межпластиночных расстояний микроструктуры $H(d_{\text{пл}})$ по формуле

$$H(d_{\text{пл}}) = H(d_{\text{пл}}^1) \cup H(d_{\text{пл}}^2) \cup \dots \cup H(d_{\text{пл}}^q),$$

где $H(d_{\text{пл}}^k) = \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} t_{ij}$;

$$\left\{ \begin{array}{l} t_{ij} = 1, \text{ если} \\ d_{ij}^{\text{пл}} \in \left((k-1) \frac{\max\{d_{ij}^{\text{пл}}\}}{q}, k \frac{\max\{d_{ij}^{\text{пл}}\}}{q} \right]; \\ \\ t_{ij} = 0, \text{ если} \\ d_{ij}^{\text{пл}} \notin \left((k-1) \frac{\max\{d_{ij}^{\text{пл}}\}}{q}, k \frac{\max\{d_{ij}^{\text{пл}}\}}{q} \right]; \end{array} \right.$$

q — заданное число интервалов; $k = \overline{1, q}$.

Потом на базе значений функции $H(d_{\text{пл}})$ составляется нормированная функция распределе-

ния межпластиночных расстояний микроструктуры $H_P(d_{\text{пл}})$ по формуле

$$H_P(d_{\text{пл}}) = H_P(d_{\text{пл}}^1) \cup H_P(d_{\text{пл}}^2) \cup \dots \cup H_P(d_{\text{пл}}^q),$$

где $H_P(d_{\text{пл}}^k) = \frac{H(d_{\text{пл}}^k)}{n}$; n — число всех видимых межпластиночных расстояний микроструктуры,

вычисляемое формуле $n = \sum_{l=1}^q H(d_{\text{пл}}^l)$.

Таким образом, используя представленный математический аппарат, можно построить функцию плотности распределения видимых межпластиночных расстояний $H_P(d_{\text{пл}})$. В качестве характеристик для параметризации микроструктуры на базе полученной функции распределения $H_P(d_{\text{пл}})$ предлагаются характеристики, представленные в табл. 1. На основе вышеизложенного математического аппарата была разработана программа для автоматизированной обработки микроструктуры перлитной стали АОМ-1 [5].

Таблица 1

Описание и формулы определения параметров микроструктур

Параметр	Наименование параметра	Формула определения
1	2	3
P_1	Истинное межпластиночное расстояние, мкм	$P_1 = \begin{cases} \frac{d_{\text{пл}}^k - d_{\text{пл}}^1}{n} s_1, & \text{если } H(d_{\text{пл}}^{k-1}) > H(d_{\text{пл}}^{k+1}); \\ \frac{d_{\text{пл}}^k + d_{\text{пл}}^1}{n} s_2, & \text{если } H(d_{\text{пл}}^{k-1}) \leq H(d_{\text{пл}}^{k+1}); \end{cases}$ <p style="text-align: right;">где $H(d_{\text{пл}}^k) = \max\{H(d_{\text{пл}})\}$;</p> $s_1 = \frac{H(d_{\text{пл}}^{k-1}) - H(d_{\text{пл}}^{k+1})}{2(H(d_{\text{пл}}^k) - H(d_{\text{пл}}^{k+1}))}; \quad s_2 = \frac{H(d_{\text{пл}}^{k+1}) - H(d_{\text{пл}}^{k-1})}{2(H(d_{\text{пл}}^k) - H(d_{\text{пл}}^{k-1}))}$
P_2	Среднее видимое межпластиночное расстояние, мкм	$P_2 = \sum_{k=1}^q H_P(d_{\text{пл}}^k) d_{\text{пл}}^k$
P_3	Доля цементитных пластин микроструктуры, %	$P_3 = \frac{\sum_{i=1}^{L_x} \sum_{j=1}^{L_y} N_{ij}^3}{L_x L_y}$
P_4	Доля видимых межпластиночных расстояний микроструктуры <0,04 мкм, %	$P_4 = H_P(d_{\text{пл}}^1)$
P_5	Доля видимых межпластиночных расстояний микроструктуры <0,08 мкм, %	$P_5 = \sum_{k=1}^2 H_P(d_{\text{пл}}^k)$
P_6	Доля видимых межпластиночных расстояний микроструктуры <0,12 мкм, %	$P_6 = \sum_{k=1}^3 H_P(d_{\text{пл}}^k)$
P_7	Доля видимых межпластиночных расстояний микроструктуры <0,16 мкм, %	$P_7 = \sum_{k=1}^4 H_P(d_{\text{пл}}^k)$
P_8	Доля видимых межпластиночных расстояний микроструктуры <0,20 мкм, %	$P_8 = \sum_{k=1}^5 H_P(d_{\text{пл}}^k)$

Параметр	Наименование параметра	Формула определения
1	2	3
P_9	Доля видимых межпластиночных расстояний микро-структуры <0,24 мкм, %	$P_9 = \sum_{k=1}^6 H_P(d_{пл}^k)$
P_{10}	Доля видимых межпластиночных расстояний микро-структуры <0,32 мкм, %	$P_{10} = \sum_{k=1}^8 H_P(d_{пл}^k)$
P_{11}	Доля видимых межпластиночных расстояний микро-структуры <0,48 мкм, %	$P_{11} = \sum_{k=1}^{12} H_P(d_{пл}^k)$
P_{12}	Доля видимых межпластиночных расстояний микро-структуры <0,64 мкм, %	$P_{12} = \sum_{k=1}^{16} H_P(d_{пл}^k)$
P_{13}	Доля видимых межпластиночных расстояний микро-структуры <1,04 мкм, %	$P_{13} = \sum_{k=1}^{26} H_P(d_{пл}^k)$
P_{14}	Минимальное видимое межпластиночное расстояние среди 10% максимальных расстояний микро-структуры, мкм	$P_{14} = d_s^z, \text{ где } d_s^z \text{ — } z\text{-й член ряда } D_n = d_s^1, d_s^2, \dots, d_s^n, \text{ который представляет собой отсортированный ряд видимых межпластиночных расстояний микро-структуры; } z = 0,9n; n = \sum_{l=1}^q H(d_{пл}^l)$
P_{15}	Среднее видимое межпластиночное расстояние среди 10% максимальных расстояний микро-структуры, мкм	$P_{15} = \frac{\sum_{l=1}^n d_s^l}{0,1n}$
P_{16}	Минимальное видимое межпластиночное расстояние среди 20% максимальных расстояний микро-структуры, мкм	$P_{16} = d_s^y, \text{ где } d_s^y \text{ — } y\text{-й член ряда } D_n; y = 0,8n$
P_{17}	Среднее видимое межпластиночное расстояние среди 20% максимальных расстояний микро-структуры, мкм	$P_{17} = \frac{\sum_{l=1}^n d_s^l}{0,2n}$
P_{18}	Высота пика функции распределения межпластиночного расстояния $H_P(d_{пл})$, %	$P_{18} = \max\{H_P(d_{пл})\}$
P_{19}	Видимое межпластиночное расстояние, соответствующее пику функции распределения $H_P(d_{пл})$, мкм	$P_{19} = d_{пл}^k, \text{ удовлетворяющее условию } H_P(d_{пл}^k) = P_{18}$
P_{20}	Площадь окрестности пика функции распределения межпластиночных расстояний $H_P(d_{пл})$, %	$P_{20} = \int_{d_1^{P_{19}-1}}^{d_1^{P_{19}+1}} H_P(d_{пл}) dH_P$

Результаты исследований

В качестве объектов для обработки изображений были использованы микро-структуры катанки перлитной стали диаметром 5,5 мм, получаемой на РУП "БМЗ". На рис. 1 представлены два образца микро-структур с различной дисперсностью, соответствующие различным плавкам из стали 80БВ.

Для каждой из этих микро-структур, используя представленный математический аппарат, были рассчитаны функции плотности распределения

$H_P(d_{пл})$, графики которых приведены на рис. 2. Как видно на рисунке, для образца № 1 с меньшей дисперсностью пик функции смещен влево, в область меньших значений расстояний по сравнению с образцом № 2, для которого характерна большая дисперсность перлита.

Для оценки устойчивости метода классификации микро-структур перлитной стали использовалась следующая методика: микро-структуры двух образцов с различной дисперсностью были условно разбиты на девять фрагментов (рис. 3, 4).

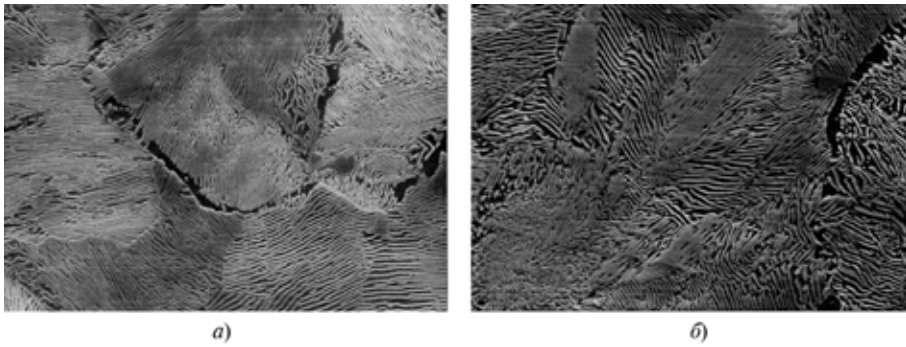


Рис. 1. Фотографии обработанных микроструктур: а — образец № 1; б — образец № 2

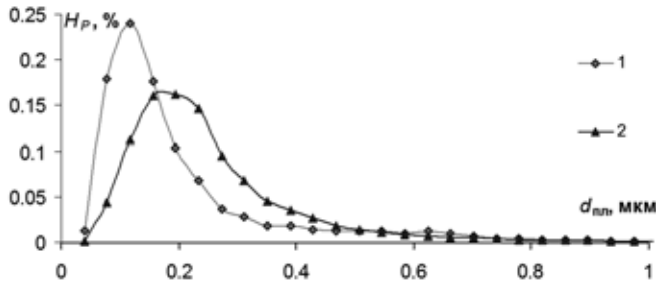


Рис. 2. Функции плотности распределения межпластиночных расстояний для двух образцов микроструктур: 1 — образец № 1; 2 — образец № 2

Обозначим фрагменты 1—9 образца № 1 как фрагменты с номерами 1—9, а фрагменты 1—9 образца № 2 — как фрагменты с номерами 10—18.

На рис. 5 представлены полученные функции плотности распределения межпластиночных расстояний $H_p(d_{пл})$ для исследованных фрагментов. Как видно на рисунке, в целом тенденция, аналогичная рис. 2, сохранилась.

На базе полученных функций распределения для фрагментов образцов микроструктур (рис. 5) была проведена параметризация фрагментов об-

разцов микроструктур (см. табл. 1), результаты которой представлены в табл. 2.

Отметим, что каждый из фрагментов, с одной стороны, был вырезан из общей микроструктуры, а с другой стороны — он отличался от общей микроструктуры, так как имел свои особенности. Как известно, сложность обработки неравновесных структур заключается в том, что они не периодичны в сечении XU . С помощью параметров табл. 1 каждый из этих фрагментов был обработан. Таким образом, каждый фрагмент характеризовался набором из 20 параметров и представлял собой таксон в двадцатимерном пространстве. На следующем этапе проводилась оценка близости фрагментов микроструктур на основе метода таксономии.

На рис. 6 представлены результаты решения задачи таксономии. Как видно на рисунке, для первого уровня близкими фрагментами являются 6-7, 15-17-18. Для второго уровня в таксон могут быть объединены 5-2-6-7, 4-8-9, 13-14. Для третьего уровня близкими являются 1-3-5-2-6-7-4-8-9, 13-14-15-17-18-10-11. Из представленных расчетов можно заметить следующую закономерность: фрагменты 1-3-5-2-6-7-4-8-9, характеризующие микроструктуру образца № 1, образуют единый таксон, что свидетельствует о близости

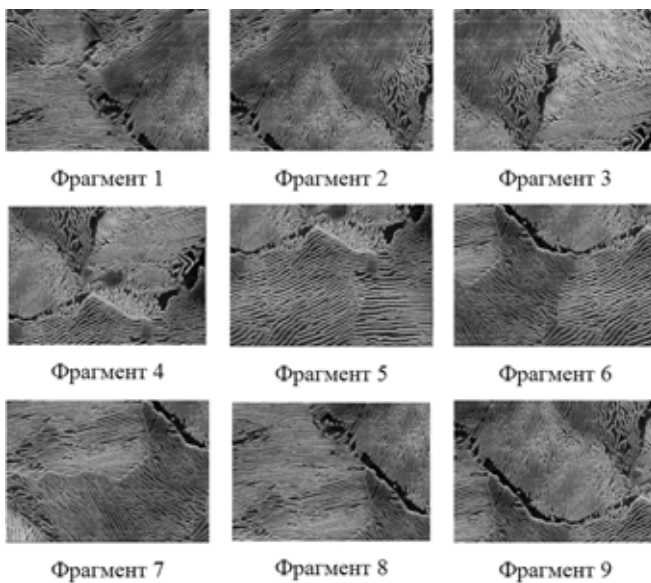


Рис. 3. Фрагменты микроструктур образца № 1

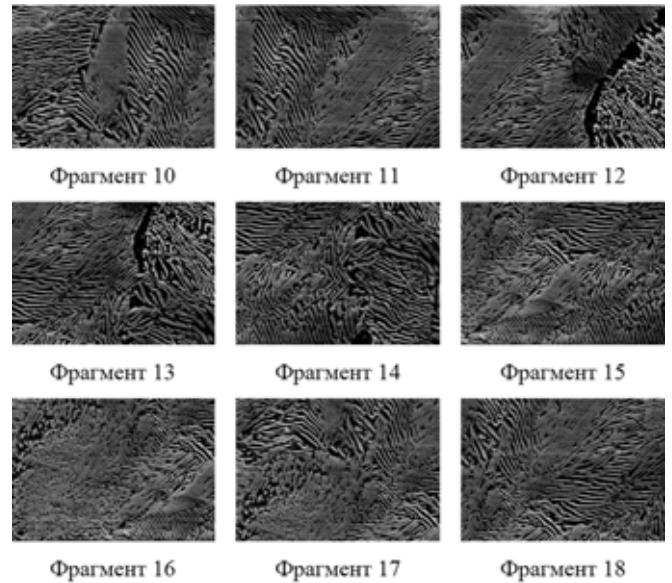


Рис. 4. Фрагменты микроструктур образца № 2

первичных фрагментов микроструктур. Фрагменты 13-14-15-17-18-10-11 также могут быть включены в единый таксон, характеризуемый параметрами таблицы. Следует отметить, что фрагменты 16 и 12 несколько отклоняются от общей закономерности, что может свидетельствовать о нетипичной микроструктуре, выбранной для исследования.

Данное исследование носит статистический характер и при использовании классификации в

задачах необходимо проводить некоторое усреднение. В целом, использование параметров функции плотности распределения показывает возможность классификации микроструктур, которые визуально плохо различимы с точки зрения дисперсности перлита. Анализируя визуально микроструктуры, представленные на рис. 4 и 5, можно видеть, что фрагменты с мелкодисперсной структурой встречаются как в первом образце, так

Таблица 2

Параметры фрагментов микроструктур для образцов № 1 и № 2

Параметр	Значения параметров изображения для девяти фрагментов образца № 1								
	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10
P_1 , мкм	0,099	0,111	0,102	0,125	0,132	0,130	0,117	0,098	0,118
P_2 , мкм	0,206	0,185	0,212	0,252	0,200	0,195	0,196	0,235	0,240
P_3 , %	0,240	0,269	0,212	0,217	0,272	0,256	0,245	0,223	0,215
P_4 , %	0,117	0,117	0,117	0,117	0,117	0,117	0,117	0,117	0,117
P_5 , %	0,612	0,646	0,565	0,541	0,638	0,615	0,615	0,578	0,532
P_6 , %	0,689	0,712	0,779	0,732	0,700	0,684	0,724	0,739	0,711
P_7 , %	0,013	0,017	0,020	0,015	0,007	0,007	0,013	0,012	0,012
P_8 , %	0,244	0,220	0,220	0,164	0,136	0,149	0,197	0,235	0,169
P_9 , %	0,484	0,490	0,433	0,380	0,408	0,405	0,442	0,458	0,384
P_{10} , %	0,624	0,663	0,585	0,557	0,645	0,622	0,628	0,591	0,544
P_{11} , %	0,709	0,758	0,674	0,652	0,782	0,761	0,750	0,663	0,631
P_{12} , %	0,766	0,817	0,741	0,719	0,861	0,848	0,828	0,714	0,696
P_{13} , %	0,839	0,885	0,828	0,785	0,900	0,899	0,890	0,779	0,777
P_{14} , мкм	0,914	0,945	0,912	0,857	0,927	0,941	0,933	0,864	0,877
P_{15} , мкм	0,962	0,981	0,968	0,911	0,953	0,971	0,967	0,939	0,951
P_{16} , мкм	0,998	1,000	1,000	0,991	0,995	0,997	0,997	0,995	0,997
P_{17} , мкм	0,468	0,351	0,390	0,585	0,351	0,351	0,351	0,546	0,546
P_{18} , %	0,312	0,234	0,273	0,351	0,234	0,234	0,234	0,351	0,351
P_{19} , мкм	0,478	0,341	0,438	0,616	0,408	0,333	0,362	0,563	0,506
P_{20} , %	0,507	0,472	0,566	0,544	0,461	0,425	0,442	0,519	0,460
Параметр	Значения параметров изображения для девяти фрагментов образца № 2								
	10	11	12	13	14	15	16	17	18
P_1 , мкм	0,217	0,175	0,164	0,186	0,219	0,176	0,144	0,177	0,184
P_2 , мкм	0,268	0,268	0,371	0,288	0,298	0,232	0,204	0,239	0,234
P_3 , %	0,178	0,174	0,16	0,151	0,158	0,179	0,210	0,165	0,185
P_4 , %	0,234	0,156	0,156	0,195	0,234	0,195	0,156	0,195	0,195
P_5 , %	0,455	0,471	0,411	0,433	0,419	0,508	0,557	0,478	0,521
P_6 , %	0,415	0,411	0,374	0,375	0,352	0,44	0,539	0,466	0,412
P_7 , %	0,001	0,002	0,001	0,001	0,001	0,003	0,005	0,002	0,002
P_8 , %	0,032	0,045	0,040	0,031	0,024	0,057	0,102	0,057	0,041
P_9 , %	0,121	0,170	0,157	0,124	0,085	0,175	0,292	0,181	0,155
P_{10} , %	0,257	0,344	0,317	0,268	0,200	0,352	0,502	0,345	0,330
P_{11} , %	0,425	0,516	0,451	0,419	0,347	0,531	0,659	0,510	0,514
P_{12} , %	0,603	0,671	0,562	0,557	0,505	0,683	0,769	0,658	0,675
P_{13} , %	0,786	0,817	0,694	0,735	0,712	0,847	0,887	0,822	0,845
P_{14} , мкм	0,920	0,914	0,808	0,889	0,886	0,963	0,972	0,952	0,965
P_{15} , мкм	0,967	0,946	0,859	0,944	0,949	0,990	0,992	0,988	0,991
P_{16} , мкм	0,997	0,978	0,927	0,992	0,995	1,000	1,000	1,000	1,000
P_{17} , мкм	0,429	0,429	0,858	0,507	0,507	0,390	0,312	0,390	0,390
P_{18} , %	0,351	0,312	0,468	0,390	0,390	0,312	0,234	0,312	0,312
P_{19} , мкм	0,487	0,490	0,935	0,541	0,507	0,403	0,270	0,376	0,365
P_{20} , %	0,587	0,499	0,484	0,525	0,507	0,501	0,473	0,532	0,491

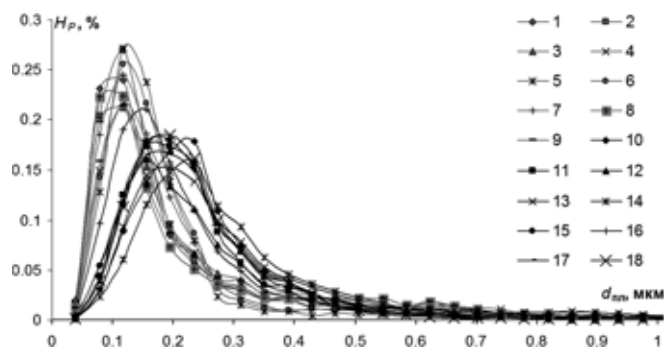


Рис. 5. Функции плотности распределения межпластиночных расстояний для 18 фрагментов двух образцов микроструктур

и во втором, но размеры этих объектов различны в представленных микроструктурах.

* * *

Таким образом, предложен метод разбиения микроструктур перлитной стали на классы, основанный на расчете функции плотности распределения по изображениям, полученным с помощью электронного микроскопа. Предложены формулы для количественного описания микроструктур, использующие интегральные зависимости для различных интервалов изменения межпластиночного расстояния. На основе метода таксономии показано, что микроструктуры могут быть разделены на два класса с помощью интегральных параметров, которые могут быть положены в основу объяснения закономерностей, формирующихся в

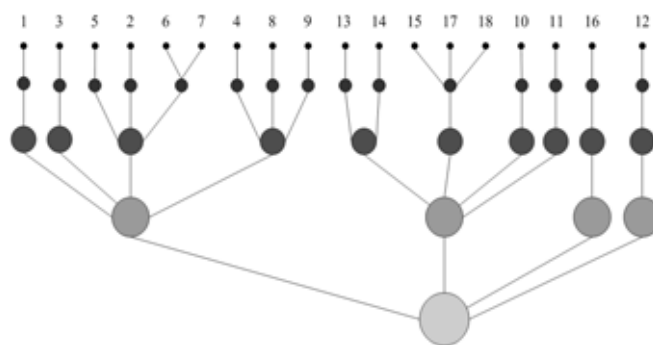


Рис. 6. Распределение таксонов по 18 фрагментам микроструктур

микроструктуре под действием факторов химического состава и технологии, что имеет важное значение для исследования взаимосвязей типа "структура — свойства" для сталей перлитного класса.

Список литературы

1. Бочвар А. А. *Металловедение*. М.: Металлургия, 1956. 494 с.
2. Уманский Я. С., Блантер М. С., Финкельштейн Б. Н. *Физическое металловедение*. М.: Металлургия, 1955. 724 с.
3. Салтыков С. А. *Стереометрическая металлография*. М.: Металлургия, 1970. 376 с.
4. Чичко А. Н., Сачек О. А., Веденев А. В., Соболев В. Ф. О новых математических методах анализа микроструктур эвтектидных колоний перлитных сталей // *Литье и металлургия*. 2008. № 4. С. 104—112.
5. Чичко А. Н., Сачек О. А., Лихозов С. Г., Веденев А. В., Бардынцева Е. П., Соболев В. Ф. Свидетельство о регистрации компьютерной программы АОМ-1 № 085. Заявка № С20090028, зарегистрирована 16.06.2009 в Национальном центре интеллектуальной собственности.

ОБМЕН ОПЫТОМ

УДК 658.012.011.56

А. В. Кузнецов, канд. техн. наук, нач. отд. ВЭД, e-mail: e-work@mail.ru,
Е. Н. Суриков, гл. технолог — помощник гл. инж. по перспективным
 и информационным технологиям, e-mail: ev_surikov@mail.ru,
 г. Рыбинск, ОАО "Конструкторское бюро "Луч"

Проблемы внедрения ERP-систем при разработке и мелкосерийном производстве высокотехнологичной продукции

Описаны особенности ERP-систем, опыт внедрения таких систем в управление производством на российском предприятии — разработчике и производителе высокотехнологичных комплексов с беспилотными летательными аппаратами и аппаратуры радиосистем передачи данных.

Ключевые слова: управление производством, ERP-система, информационные технологии, сопровождение продукта, жизненный цикл продукта

Современные ERP-системы, содержащие производственные модули ("BAAN", "SAP R3", "Ехас", "MGF/PRO", "Галактика", "БОСС-Корпорация"), решают сложные задачи управления финансовыми и материальными пото-

ками на верхнем уровне предприятия (сбыт, снабжение, склады, бухгалтерия и т. п.), но, увы, не на уровне цеха и, уж тем более, участка. Другими словами ERP-системы решают глобальные задачи управления предприятием

или объединением в целом, а что творится на производственном участке — практически не является сферой действия ERP-системы, важен только конечный результат деятельности этого участка. А между тем, чем ближе предприятие с точки зрения классификации масштабов производства находится к мелкосерийному производству, тем ощутимее его потери вследствие отсутствия контроля и управления, прежде всего на цеховом уровне, где реализуются основные производственные процессы. Во-первых, сюда следует отнести конструкторско-технологическую подготовку производства с использованием САД-систем для выполнения конструкторских работ, САМ-систем для подготовки управляющих программ для оборудования с ЧПУ и технологических систем для разработки технологий изготовления изделий на имеющемся универсальном или специализированном оборудовании. Во-вторых, сюда следует отнести решение задачи организации производства.

В настоящее время бытует ошибочное мнение, что для решения задач организации производства более чем достаточно использовать любую из ERP-систем, представленных на современном рынке программного обеспечения. Но весь фокус заключается в том, что организация производства предполагает, в частности, и управление производственными процессами на уровне цеха, участка, включая внутрицеховые склады. А здесь, как уже отмечалось ранее, ERP-система на сегодняшний день не работает.

Фактически на производственном участке завода может находиться большое число станков, которые необходимо загрузить с максимальной эффективностью (прибылью) для предприятия. В то же время, одной из существенных проблем единичного и мелкосерийного производств является низкий коэффициент загрузки оборудования. По данным Международной организации инженеров-технологов средний коэффициент загрузки оборудования мелкосерийных и единичных производств составляет 0,45. Другими словами, больше половины рабочего времени современное дорогостоящее оборудование стоит на производственных площадях "мертвым капиталом", не принося прибыль и морально устаревающая. Этот показатель одинаково справедлив для предприятий в любой точке мира и неизбежно остается таковым, если не применять специальных методов, способствующих более полной загрузке оборудования. Причина неэффективной загрузки цехового оборудования кроется в самой сути единичного и мелкосерийного производства.

Известно, что тип производства определяется не столько размером обрабатываемых партий деталей, сколько так называемой плано-учетной единицей, используемой для его управления.

Если производственное подразделение работает и отчитывается по заказам (заказ — это совокупность деталей собственного изготовления, которые впоследствии собираются в изделие), то плано-учетной единицей выступает производственный заказ, а само производство относится к единичному или мелкосерийному. Для позаказного производства определяющим фактором является комплектность выпускаемого изделия. Если хотя бы одна деталесборочная единица не изготовлена, все остальные детали, составляющие заказ, лежат мертвым грузом, образуя незавершенное производство, и весь заказ считается невыполненным. Таких заказов в мелкосерийном производстве может быть тысячи, и, например, одна и та же втулка (один типоразмер) может входить одновременно сразу в несколько заказов. Но изготавливать все эти втулки будут в разное время и, возмож-

но, на разном оборудовании, так как каждый из заказов рассматривается как единый комплект деталесборочных единиц (согласно международному стандарту ISO-9000 детали различных заказов должны быть идентифицируемы, т. е. их нельзя смешивать).

Одним из основных путей повышения коэффициента загрузки оборудования является составление производственного расписания его работы, контроль его выполнения и оперативное внесение в расписание корректив.

На практике в производстве без завершения обработки всех деталей, входящих в узел, не начнется и сборка этого узла, или при задержке одной технологической операции невозможно приступить к следующей. В результате даже небольших отклонений от производственного расписания начнутся простои основного технологического оборудования, часть деталей будет пролеживать, ожидая в очереди на обработку, необоснованно возрастет объем незавершенного производства.

Правильно составить расписание загрузки оборудования — это решение только части задачи. Нормы времени, которые вносит технолог, даже по результатам расчетов в хорошо зарекомендовавших себя технологических системах, на практике, как правило, не выдерживаются. Более того, можно составить управляющую программу для станка с ЧПУ и по ней рассчитать нормы времени для отдельных технологических переходов, но эти данные скорее всего окажутся неточными. Действительно, скорость подачи инструмента на управляющей стойке станка, исходя из реальных условий (качества режущего инструмента, состояния поверхностного слоя материала) в конечном итоге устанавливает сам оператор, главной задачей которого является, прежде всего, получение точных и качественных поверхностей обрабатываемой детали. В результате, отклонение от норматива может составить в среднем 20 % как в сторону увеличения времени обработки, так и в сторону его уменьшения. И соответственно, составленное исходное расписание будет отличаться от фактического. Другими словами, ни одно производственное расписание в цехе реально не выполняется.

На практике принято считать, что сбой в производственном расписании начнут проявляться уже через два часа после начала его выполнения. Ситуацию наглядно описали американские специалисты в области управления производством Р. Сури (R. Suro) и С. Уитни (С. Whitney): "Сбой могут расширяться, как пожар, и тогда управляющему (участком, цехом) приходится работать как пожарному". Суть же проблемы четко сформулировал С. Уайт (S. White) — классик производственного планирования: "Календарные планы производства требуют корректировки сразу после их составления".

Практика показывает, что если не осуществлять компьютерную поддержку производственного расписания в процессе его выполнения, то ожидать существенного увеличения коэффициента загрузки оборудования не приходится.

Составление оптимального расписания загрузки оборудования, диспетчерский контроль его выполнения, а также внесение своевременных корректив в текущий производственный план являются важными задачами организации производства. В то же время, для организации эффективного управления на уровне цеха эти задачи неотрывно связаны с организацией учета материальных ценностей на внутрицеховом складе и с контролем их движения между складом и производственными участками цеха.

Итак, современное программное обеспечение автоматизации управленческой и инженерной деятельности

промышленных производств условно можно разделить на три информационно взаимосвязанные группы:

- системы для управления предприятием на верхнем уровне;
- системы для внутрицехового управления;
- системы для конструкторско-технологической подготовки производства.

В идеале первые две группы должны быть объединены в одну, но на сегодняшний день ERP-системы, представляющие первую группу, практически не решают задач управления производством на уровне цеха и участка единичного и мелкосерийного производства, что привело к появлению нового класса компьютерных программ, решающих эти задачи и обеспечивающих гармоничную связь между системами конструкторско-технологической подготовки производства и ERP-системами.

Применительно к ОАО "КБ "Луч" внедрение ERP-системы было рассмотрено не как комплекс отдельных мероприятий по использованию информационных технологий в управлении, а как разработка комплексной стратегической программы развития информационных технологий и ее привязки к общей бизнес-стратегии предприятия.

Технические средства на предприятии представлены в виде персональных компьютеров числом более 500 шт., объединенных между собой единой вычислительной сетью.

На предприятии создан механизм, позволяющий описывать данные об изделии на протяжении всего жизненного цикла. С этой целью внедряется отечественная система ЛОЦМАН фирмы АСКОН, которая позволит создать единое информационное пространство, базу данных об изделиях коллективного пользования и подготавливать электронно-технические руководства для выпускаемых изделий.

Система ЛОЦМАН содержит всю информацию, необходимую для проектирования, изготовления и эксплуатации продукции.

На этапе подготовки производства система обеспечивает накопление данных о результатах конструкторско-

технологического проектирования и обмен информацией между инженерными службами.

Разработчики радиоэлектронной аппаратуры в тематических подразделениях используют в работе средства схемотехнического анализа и моделирования "OrCad, PCAD, Max+, для разработки СВЧ узлов используется Microwave Office.

В конструкторском отделе применяются системы, автоматизирующие процесс создания механических конструкций и моделирования трехмерных моделей Solid-Work, Компас-3D, AutoCad, для печатных плат используется PCAD.

Технологи разрабатывают управляющие программы для станков с ЧПУ с помощью систем Компас-ЧПУ и Gemma-3D.

Подготовка производства, составление планов загрузки цехов проводится на основе единых данных в системе ЛОЦМАН:PLM.

Бухгалтерский учет ведется с помощью системы 1С: Предприятие и программных средств собственной разработки.

Дальнейшее развитие информационных технологий на предприятии позволит значительно повысить качество разрабатываемой аппаратуры и комплексов в целом, а также сократить сроки разработки, следовательно, снизить издержки на производство и повысить конкурентоспособность предприятия. Руководство предприятия понимает, что финансирование информационных технологий — это инвестиции в будущее предприятия.

Список литературы

1. О'Лири Д. ERP-системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение, эксплуатация / Пер. с англ. Ю. И. Водяновой. — М.: Вершина, 2004. 272 с.
2. Козловский В. А., Козловская Э. А., Савруков Н. Т. Логистический менеджмент. СПб.: Лань, 2002. 272 с.
3. Филипс Д. Менеджмент ИТ-проектов. На пути от старта до финиша. М.: ЛОРИ, 2005. 375 с.
4. Севрюгин Н. Н., Юдин А. В., Кузнецов А. В. О методологии выбора технических решений // Автоматизация и современные технологии. 2005. № 3. С. 31—37.

CONTENTS

Krivtsov O. A., Korikov A. M. Methods of Tracking Head in a Video Sequence: State of the Problem	2
The main application areas of human's head tracking are reviewed; challenges arising in head tracking are listed. A survey of historical evolution of head tracking methods and the difficulties arising in the field of head tracking are presented.	
Keywords: head tracking, video frames, tracking methods, historical evolution, tracking difficulties	
Vahromtsev E. M. "Live" Modelling Methods Application for a Business Monitoring.	10
The article discovers the unique way for a fast diagnostic and monitoring of the control process with a help of the "live" modeling technique including a logical neural network and a modern implementation of the skeletal animation. The "live" modeling allows to show current business processes in a form of animated 3D virtual character.	
Keywords: "live" modeling, virtual person, skeletal animation, computer graphics, monitoring, fast diagnostic, logical neural network	
Zhuk D. M., Manichev V. B., Ilitskiy A. O. Methods and Algorithms of the Differential-Algebraic Equations Solution for Simulation of Systems and Objects in the Time Domain. Part 1.	16
In part 1 of this article results are presented for the analysis and explanation of the choice of a base methods for the solution of nonlinear systems of differential-algebraic equations (DAE) for the purpose of development of the new universal, highly reliable software suite FMS-PA10 (Framework for Mathematical Simulation, Program for Analysis version 10) for systems and objects simulation in the time domain. The new combined adjusting have been developed for realization of AL-steady integration methods of DAE systems solution (on the example of an implicit trapezoidal integration method). This algorithm will be used for the development of software suite FMS-PA10.	
Keywords: simulation, Ordinary Differential Equations (ODE), Differential Algebraic Equations (DAE), integration methods, time domain, Linear Algebraic Equations (LAE)	
Perepelkin E. A., Poddubnyi V. I., Warkentin A., Hahn M. Application of Applied Package CAMEL-View for Modelling Controlled Movement of a Wheel Tractor.	24
In article the brief description of the basic objects of applied package CAMEL-View (Germany) for creation mechatronic models of mechanical systems and its opportunities are presented. The model of a wheel tractor with jointed frame, created with use CAMEL-View is considered. The synthesis algorithm of the PI-regulator with the observer in feedback and the control system	

developed on its basis of movement of a wheel tractor are presented. The results of modeling of responses in MATLAB and set movement of a tractor in CAMEL-View confirm adequacy of its model and a control system of movement.

Keywords: control, mathematical model, wheel tractor, state space, proportional-plus-integral action controller

Kosnikov Yu. N. Terrain Geometric Simulation and Displaying on Topo Data 30

The terrain display technology on a physical geographical map is offered. Stages of technology: map raster image obtaining, topo data rebuilding, creation of a surface geometrical model, tessellation, shading, texture mapping.

Keywords: map scanning, image filtering, scalar field, surface geometrical model, tessellation, shading

Nemtinov V. A., Gorelov A. A., Nemtinov K. V., Kulin D. V. Spatial-Time Model of an Object of Cultural and Historical Heritage . . . 36

The article examines the model of information object of cultural-historical value, presented in the form of graphs structures of frames and including data on structure, properties of the object and its elements, and also ways of these values input, allowing an expert to systematize information on real physical object, to order its storage on electronic carriers and to provide its effective processing.

Keywords: object of cultural-historical value, information object, analytical models, information and logical models

Bogdanov M. R., Bogdanova A. V., Gabidullin Yu. Z. Wireless Sensor Networks in Ecology 39

The review considers the principles of wireless sensor networks (WSN). Examples of the WSN in environmental are considering. Some problems arise when using WSN in environmental studies. Consider the current and future sensors.

Keywords: wireless sensor networks, ecology

Osipov V. Yu. Recurrent Neural Network with Operated Synapses 43

The recurrent neural network controlled synapses in possession of the advanced facilities by intelligent information processing is offered. The paper discloses facilities for realization in it of the processing signal controlled spatial shifts. The information storage rules on the neural synapses are specified. Results analysis of computing experiment is gave.

Keywords: the recurrent neural network, operated synapses, structure, rules, intelligent information processing controlled

Kryuchin O. V. Neurons's Network's Simulator with Automatic Selection of Activations Functions 47

Article describes algorithm of neurons's networks's teaching with automatic selection of activations functions. Program with this algorithms realisation and ways of speeding are written too.

Keywords: neural networks, parallel algorithms

Zhukov O. D. Information-Computational Technology Based on Polynomial Algebra 52

The paper is intended for a review description of polynomial remainder system that is similar to remainder number system. Its main difference is a simpler polynomial division. The discussed system is efficient for polynomial computation.

Keywords: remainder system, polynomial algebra, interpolation polynomial

Kalitin D. V. Design of Parallel Decomposition of Automata in k-Valued Logics. 60

The paper considers the design strategy of parallel decomposition of automata. This strategy is based on theoretical and graph embedding graph clutch automaton graph of special form — a cluster of functional incoherence. As a result of such decomposition is possible to design a function that will describe the work of these machines. Further, the obtained functions are implemented using digital 3-valued neurons. The presented strategy is immune from deficiencies of methods previously used. Also, the proposed method allows you to receive automatic factors at once, i. e. parallel, not sequentially as previously.

Keywords: finite automaton, decomposition, graph

Mikheeva T. I., Saprykin O. N. Design Patterns of Analytical Subsystem of Spatial Objects State of Intelligent Transportation Systems . . . 64

In article requirements with which the intelligent transportation system analyzing toolbox should satisfy are defined. Use of design patterns is chosen as a way of satisfaction to requirements. Design patterns of different levels are considered and the way of their application in intelligent transportation system is shown.

Keywords: geoinformation system, class, object, client, server, service-oriented design

Chichko A. N., Sachek O. A., Likhovozov S. G. Methods of Automation of Processing of Images of Microstructures of Pearlite Steels 71

Mathematical methods for processing of images of microstructures pearlitic steels are described. Methods allow to conduct quantitative and qualitative analysis of characteristics of a microstructure on a basis of distances between pearlite of colonies of phases. Methods are based on approximation of a microstructure by functions of density of distribution of distances between pearlite of colonies. It is shown that microstructures pearlitic steels can be divided into two classes on the basis of distribution functions of distances between pearlite of colonies.

Keywords: processing of images, microstructures, pearlitic steel, distribution function of distances between pearlite of colonies, dispersivity of pearlite

Kuznetsov A. V., Surikov E. N. Problems of ERP-Systems Introduction at Development and Small-Scale Manufacture of HI-TECH Production 77

Article is devoted to features of ERP-systems, experience of introduction of such systems in production management at Russian company — developer and manufacturer of hi-tech systems with unmanned aerial vehicles and equipments of datalink radiosystems.

Keywords: production management, ERP-system, information technologies, product support, product life cycle

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер Т.Н. Погорелова. Технический редактор О. А. Ефремова.

Корректор Е. В. Камиссарова

Сдано в набор 05.05.2010. Подписано в печать 17.06.2010. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 11,63. Заказ 549. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,

телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"

142110, Московская обл., г. Подольск, ул. Кирова, 15