

НАУЧНО • ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ
ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г.

№ 12

Москва 2012

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.81

Е.Ф. Фабрикантова

Моделирование систем средствами искусственного интеллекта*

Излагаются соображения по методике исследования, предпринимаемого для выявления связи между психическими расстройствами человека и функциональным кожным зудом. Предлагается гипотеза об образовании в организме больного патологической системы (и заболевание, и систему мы условно назвали «Психозуд»). Для проверки гипотезы предлагается создать модель этой системы с помощью методов искусственного интеллекта. Приводятся некоторые сведения из теории систем. Рассматриваются вопросы многоагентного моделирования систем. Обсуждаются возможности использования интеллектуального анализа данных на основе ДСМ-метода для автоматического порождения закономерностей поведения системы «Психозуд».

Ключевые слова: система, патологическая система организма, моделирование систем, искусственный интеллект, многоагентная система, интеллектуальный анализ данных, ДСМ-метод автоматического порождения гипотез, интеллектуальная система типа ДСМ

ВВЕДЕНИЕ

Задача исследования заболевания, при котором у больных существует тесная связь между психическими расстройствами и кожным зудом («Психозуд»)

была поставлена сотрудниками Научного центра психического здоровья (НЦПЗ) РАМН [1]. Углубление в предметную область привело к пониманию того, что для решения этой задачи необходимо:

1) рассматривать данное заболевание как возникновение в организме человека патологической системы (гипотеза);

* Работа выполнена при поддержке РФФИ (проект № 11-07-00618а) и Президиума РАН (П15, проект № 205).

2) использовать для проверки гипотезы, т.е. при компьютерном исследовании и моделировании системы «Психозуд», методы искусственного интеллекта (ИИ) как наиболее адекватные задаче.

Дадим некоторые пояснения.

Если рассматривать данное заболевание как эмерджентное (emergent – появляющийся) поведение организма, то следует принять гипотезу об образовании в нем патологической системы, в которую, по крайней мере, должны входить мозг и кожа (конечно, при подтверждении гипотезы о выявлении связи между расстройствами психики и кожным зудом у больных).

Необходимость создания модели системы очевидна: именно моделирование является одним из способов проверки правильности нашего понимания явлений и процессов. Компьютерное же моделирование в медицине особенно востребовано в силу проблем (в первую очередь, этических), связанных с проведением клинических исследований, а также с экономическими и временными затратами на эксперименты в лабораториях.

О выборе методов ИИ скажем следующее. Существующие методы моделирования систем (математические и имитационные, использующие статистические методы) не вполне удовлетворяют исследователей [2]. Дело в том, что использование математических моделей (в основном дифференциальных уравнений) имеет смысл для не очень сложных систем. Более сложные системы описываются дифференциальными уравнениями высоких порядков, для которых не существует (по крайней мере, в настоящее время) методов для аналитического решения. Конечно, существуют численные методы решения таких уравнений. Однако экономические, биологические системы описываются в основном не числовыми, а качественными характеристиками. Именно так обстоит дело в рассматриваемой задаче: данные для описания больных представляют собой, в основном, феноменологические, качественные признаки, объективных (числовых) параметров мало.

Что касается использования имитационных статистических методов, то в рассматриваемой задаче, как и в большинстве задач медицины, процессы считаются детерминированными, а, следовательно, эти методы для медицинских задач неадекватны. Отметим еще, что в имитационных моделях отсутствует явное представление знаний для оценки, анализа и интерпретации результатов и самого процесса имитации.

ИИ всегда был нацелен на решение таких задач. Развитие агентного подхода в ИИ и, особенно, многоагентных систем значительно расширило область применения ИИ. Создание моделей для исследования систем находится в настоящее время на переднем крае научных и технологических разработок в области ИИ. Используя компьютерное моделирование на основе агентов, можно исследовать сложные системы, для которых не существует аналитического описания и которые из-за этого ранее были недоступны для изучения. Методы агентного моделирования применяются для описания широкого спектра явлений, например, адаптации человеческой иммунной системы, управления сложными процессами, включая ускорение частиц, поведения мировых рынков валют, метеорологии. Многие исследователи системного моделирования занимаются созданием моделей

для стаи птиц, роя пчел, муравьиных куч, которые являются природными примерами порождения эмерджентного поведения и многоагентных систем [2, 3]. Было выявлено много важных примеров эмерджентного поведения и созданы приложения, предназначенные для управления городами, руководства предприятиями, а также для использования в других весьма разнообразных областях [4].

Теория систем и многоагентное моделирование ИИ взаимно обогатили друг друга. Теория систем дала ИИ огромный класс объектов для исследования, а также стимул для развития и демонстрации своих возможностей при решении многих важных задач. ИИ предоставил исследованию систем новые модели, переход к использованию качественных характеристик для описания поведения системы и ее элементов, а также средства представления знаний о поведении и, тем самым, значительно расширил круг системных исследований.

Конечно же, выявление знаний о поведении агентов и их взаимодействии – чрезвычайно трудная, творческая задача. Чаще всего в начале исследования выдвигается гипотеза о существовании системы, иногда задается ее интегративное свойство, общие характеристики, описывающие систему, – все остальное требуется установить в ходе исследования – и, в первую очередь, какие именно элементы входят в систему.

Предварительному исследованию гипотезы о системе «Психозуд» может помочь автоматическое порождение закономерностей в результате интеллектуального анализа данных, реализуемое ДСМ-методом автоматического порождения гипотез [5, 6]. Этот метод относится к активно развивающемуся в настоящее время и широко используемому направлению ИИ Data Mining [7], а точнее, к Knowledge Discovery – автоматическому порождению знаний.

Таким образом, предлагаемое исследование должно состоять из двух частей:

1. Анализ данных для получения метаинформации: требуется установить, действительно ли образовалась система, определить ее элементы, выявить системообразующие связи и потоки ресурсов, идущих по этим связям.

2. Создание первоначальной компьютерной модели системы с использованием результатов анализа данных, проведение экспериментов, уточнение модели и т.д. (реализация итерационного процесса).

В разделе 1 статьи рассматриваются понятия «система» и «интегративное свойство», приводятся необходимые краткие сведения из теории систем [8-12].

В разделе 2 описывается компьютерное моделирование, как оно рассматривается в ИИ, и, в частности, компьютерное моделирование систем в виде многоагентных объектов. Приводится пример порождения эмерджентного поведения стаи птиц [3]. Обсуждаются понятия «эмерджентное поведение» и «эмерджентность».

В разделе 3 рассматриваются предлагаемая гипотеза о системе «Психозуд» и вопросы создания модели этой системы.

Раздел 4 содержит краткое описание целей и ожиданий от интеллектуального анализа данных о систе-

ме «Психозуд» с помощью ДСМ-метода автоматического порождения гипотез. Приводятся соображения о возможности использования интеллектуальной системы типа ДСМ (ИС-ДСМ) в качестве инструмента для моделирования.

Итак, в настоящей статье рассматриваются методы искусственного интеллекта, с помощью которых можно моделировать и исследовать системы. Обсуждаются вопросы их применения для создания модели патологической системы организма «Психозуд».

1. НЕКОТОРЫЕ СВЕДЕНИЯ ИЗ «КЛАССИЧЕСКОЙ» ТЕОРИИ СИСТЕМ

1.1. О понятиях «система» и «интегративное свойство»

Какой сложный объект (состоящий из нескольких элементов) можно считать системой?

Определение из словаря: «система (греч. *systema* (целое), составленное из частей; соединение) – множество закономерно связанных друг с другом элементов (предметов, явлений, взглядов, знаний и т.д.), представляющее собой определенное целостное образование, единство».

Одно из научных определений системы следующее: «Если некоторая совокупность элементов обладает свойством, которое не является суммой или средневзвешенным свойств отдельных элементов, то такое свойство называется интегративным (эмерджентным), а сама совокупность называется системой» [8]. Термины «сумма» и «средневзвешенное» понимаются как метафоры, так как они возникли как дань математическим методам моделирования и исследования систем в классической теории систем.

Невозможно свести свойства системы к сумме свойств ее компонентов. Именно поэтому нельзя просто перейти от молекулярной биологии к биологии клеток, организмов, популяций и т.д. И наоборот, нельзя свести психологические и социологические явления к биологии. Сложное = сумма простого + взаимодействие. Именно взаимодействие изменяет свойства элементов. Каждый из элементов системы утрачивает индивидуальность в результате взаимодействия элементов, образования системных связей между ними. У них уменьшаются степени свободы. Системный анализ [9], исследование систем – это, прежде всего, анализ связей: в физическом мире анализ взаимодействий, в математике – отношений, вообще в науке – взаимных влияний объектов [12].

Итак, система – это целостное образование, и ее поведение имеет закономерности, отличные от закономерностей поведения ее элементов.

Примеры систем и их интегративных свойств. Свойство атома – образовывать связи определенной энергии с другими атомами, причем энергия связей между атомами не является ни суммой, ни средним энергия связи между элементарными частицами.

Свойство дерева – образовывать семена, из которых могут вырасти новые деревья, что не является суммой свойств отдельных стволов, листьев, корней и т.д.

Смысл предложения не является суммой смыслов входящих в него слов.

В физике эмерджентные феномены: трение, вязкость, эластичность, температура, конвекция; явления природы – ураганы и погода вообще. В биологии – это комбинирование молекул в полипептиды, которые, взаимодействуя, самоорганизуются в органеллы, далее в клетки, ткани, органы, организмы; склонность живых существ к объединению в стаи.

1.2. Некоторые сведения из теории систем

Образование системы (о естественных, природных системах говорят «самоорганизация») – это творческий акт.

По определению, чтобы совокупность стала системой, у нее должно возникнуть интегративное свойство. Для этого компоненты совокупности должны стать определенным образом взаимосвязанными, и по ним должен пойти поток определенного ресурса. Система, лишенная на какое-то время ресурса, потенциально остается системой, только ее интегративное свойство не проявляется, остается потенциальным. А вот система, лишенная структуры, теряет интегративное свойство, даже если ресурса у нее в достатке. Автомобиль со снятыми колесами не поедет никуда, даже если его бак целиком залить бензином [8]. Отсюда вывод: для разрушения системы можно нарушить ее структуру, например, разорвать какие-то связи между элементами.

Естественные системы представляют собой псевдостатические объекты. Это означает, что элементы системы реализуют динамическое поведение, а сама система – статическое, например, как стая птиц (см. об этом в разделе 2).

В системе реализуется децентрализованное распределенное управление.

Понятие цели системы, предлагаемое в работах П.К. Анохина (см., например, [10]), часто отождествляют с интегративным свойством [8].

2.2. О КОМПЬЮТЕРНОМ МОДЕЛИРОВАНИИ СИСТЕМ С ПОМОЩЬЮ АГЕНТНОГО ПОДХОДА

2.1. Модель, теория, знания

В математике понятия «модель» и «теория» имеют строгие определения. В XX в. понятие модели, например, становилось все более общим, охватывающим и реальные, и идеальные модели. При этом понятие абстрактной модели вышло за пределы математических моделей, стало относиться к любым знаниям и представлениям о мире. (Следует отметить, что споры вокруг такого широкого толкования понятия модели продолжаются и поныне).

В ИИ понятия «знания», «модель» и «теория» чаще всего используются в их широком общенаучном смысле, а в настоящее время еще и как синонимы. Это связано с тем, что ИИ занимается предметными областями, в которых процесс познания еще не завершен, так называемыми открытыми предметными областями. Для них нельзя создать аксиоматические теории. Моделирование для этих областей и применяется с целью создания теорий.

В подтверждение синонимичности использования понятий модель, теория и знания приведем несколько распространенных определений и цитат.

Под моделью некоторого объекта понимается другой объект (реальный, знаковый или воображаемый), отличный от исходного, который обладает существенными для целей моделирования свойствами и в рамках этих целей полностью заменяет исходный объект.

«В сфере научных дисциплин информационного, кибернетического, системного направлений слово «модель» употребляют для обозначения модели теории, которая описывает класс наблюдаемых объектов. Например, когда демонстрируется движущаяся модель черепахи в виде тележки на колесах с мотором, то это, строго говоря, не модель самой черепахи, а модель той теории, которая описывает класс объектов, способных совершать простые движения и выполнять несложный набор команд. Точно так же, когда говорят, что так называемая нейронная сеть (т. е. множество простых электронных элементов с четко определенной системой связей и логикой действий) есть модель мозга, то это надо понимать как модель некоторого очень грубого представления о том, как может быть устроен мозг». [13].

«Программы могут быть кристально ясным, четким описанием теорий. ... Часто после знакомства с тем, что может делать программа, спрашивают, а какая теория стоит за ней. Подчас правильным ответом будет, что эта программа и есть теория» [14].

«Знания о том, «как работает мир» (которые могут быть воплощены в простых логических схемах или сложных научных теориях) называются моделью мира» [3].

Теория – одна из форм представления знаний, их организация.

Можно считать, что в ИИ модель некоторого реального объекта есть теория этого объекта, которая может быть описана, в частности, в виде знаний, представленных с помощью некоторой схемы (как принято говорить в ИИ, с помощью «модели представления знаний»: семантической сети, онтологии, фрейма, продукционной системы и т.д.).

2.2. Моделирование систем с помощью агентного подхода искусственного интеллекта: пример модели системы «стая птиц»

В литературе можно найти множество различных определений агентного моделирования. С точки зрения практического применения агентное моделирование можно определить как метод имитационного моделирования, исследующий поведение децентрализованных агентов и то, как это поведение определяет поведение всей системы в целом. При разработке агентной модели разработчик вводит параметры агентов (агентами могут быть люди, компании, активы, проекты, транспортные средства, города, животные и т.д.), определяет их поведение, помещает их в некую окружающую среду, устанавливает возможные связи, после чего запускает моделирование. Индиви-

дуальное поведение каждого агента образует глобальное поведение моделируемой системы.

«Агентные модели оказали немалое влияние на теорию сознания. Например, Даниэль Дэннет рассматривает функцию и структуру сознания, отталкиваясь от агентной архитектуры интеллекта. Его множественная теория сознания основана на рассмотрении сознания во взаимодействии агентов в распределенной архитектуре интеллекта. Во время перцепции, управления двигательными функциями, решения задач, обучения и другой психической активности формируются объединения взаимодействующих агентов. Эти сочетания очень динамичны и изменяются в зависимости от потребностей в различных ситуациях. Сознание, по Дэннету, служит связующим механизмом этих объединений агентов, оно поддерживает взаимодействие агентов. Совокупность агентов становится основой когнитивной обработки данных. ...Интеллект подобен фрактальной структуре, где одни и те же процессы действуют в любых масштабах и во всей системе в целом» [2].

Моделирование систем – частный случай агентного моделирования. Модели систем так же, как их прототипы, должны демонстрировать эмерджентное поведение. У естественных систем, рассматриваемых в данной работе, эмерджентное поведение – это образование некоторого псевдостатического объекта из множества динамических элементов.

Рассмотрим пример – как ведут себя птицы, сбиваясь в стаи во время полета [3].

«Вполне приемлемая модель такого процесса может быть получена, если каждый птицеподобный агент использует три перечисленных ниже правила, применяя определенный метод их комбинирования.

1. **Отделение.** Направлять свой полет подальше от соседей, если ты начинаешь к ним слишком заметно приближаться.

2. **Соединение в линию.** Придерживаться направления полета, позволяющего занять среднюю позицию по отношению к соседям.

3. **Выравнивание.** Рулить в сторону средней ориентации (направления движения) соседей.

Если все птицы выполняют одни и те же описанные выше правила, то вся стая обнаруживает эмерджентное поведение (от слова *emergent* – появляющийся), совершая полет в виде одного псевдоустойчивого тела приблизительно с постоянной плотностью, которое не рассеивается со временем. Как и в случае общественных насекомых, для каждого агента не требуется обладание совместным планом, который моделировал бы действия других агентов» [3].

Заметим, что эти правила являются конкретизацией для данного случая универсальных правил поведения элементов в системе: следи за соседом, не приближайся к нему, придерживайся «середины» и т.д. [4].

Итак, многоагентная система состоит из элементов. Агенты имитируют структурные элементы изучаемого объекта, процесса или явления. Знания, представленные в многоагентной системе, должны отражать поведение каждого элемента системы, их взаимодействие и эмерджентное свойство. Взаимо-

действие агентов моделирует динамику изучаемого объекта, процесса или явления. В природных системах эмерджентное (интегративное) поведение – это образование псевдостатического объекта. Чаще всего знания представляются в виде правил. В рассмотренном примере правила, задающие поведение каждой птицы-агента, – это правила динамического поведения птицы в стае.

2.3. Об эмерджентном поведении и эмерджентности

В настоящее время при исследовании систем, наряду с интегративным, эмерджентным свойством, эмерджентностью, рассматривается еще и поведение системы и ее элементов. Это пришло из ИИ [15], теории управления, кибернетики [16]. Следовательно, можно дать следующее определение системы: если поведение некоторой совокупности элементов отличается от поведения каждого из них, то они образуют систему. Или: интегративное, эмерджентное поведение характеризует совместное поведение объектов, но не относится ни к одному из них в отдельности.

Однако и здесь можно говорить о расширении использования термина «эмерджентное поведение». Приведем цитату.

«Как эмерджентное характеризуется неожиданно возникающее поведение, которое невозможно было предвидеть заранее. Например, может показаться, что вода в ручье течет очень плавно, но это впечатление внезапно нарушается после падения в воду камня. В зависимости от скорости воды, размера камня и других факторов может возникнуть турбулентность. Вода обнаруживает нелинейное поведение в случае газифицированного потока, разрывного течения и в других случаях, в которых вступают в силу динамические изменения. Эти изменения не могут быть предсказаны с помощью обычной гидродинамики и требуют применения для своего описания дифференциальных уравнений второго или более высокого порядка. Чаще всего получение аналитического решения этих уравнений не представляется возможным, поэтому самый лучший способ действия, который возможен в таких случаях, состоит в получении приближенных числовых решений» [17].

Здесь как об эмерджентном говорится о поведении воды, которая является элементом системы «вода – камень», а не всей системы в целом.

По-видимому, следует различать термины «эмерджентное поведение» и «эмерджентность». Эмерджентность означает вообще наличие у некоторой системы свойств, не присущих ее элементам, и даже не присущих их сумме, т.е. термин «эмерджентность» означает наличие «системного эффекта», интегративное свойство и относится ко всей системе, а эмерджентное поведение может относиться как к системе, так и к ее элементам. Итак, можно сказать, что эмерджентное поведение стаи птиц, ее эмерджентное свойство – образование псевдостатического тела, а эмерджентное поведение птицы в стае – это полет по вышеприведенным правилам, который отличается от полета одинокой птицы.

3. ГИПОТЕЗА О ПАТОЛОГИЧЕСКОЙ СИСТЕМЕ ОРГАНИЗМА «ПСИХОЗУД» И СОЗДАНИЕ ЕЕ МОДЕЛИ СРЕДСТВАМИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

3.1. Гипотеза

Предлагаемая гипотеза «Психозуд» состоит в следующем. Нормальной работой системы «мозг – кожа» является восприятие тактильных воздействий на организм, т.е. обеспечение одного из пяти каналов связи организма с окружающей средой – тактильного (наряду со зрительным, слуховым, обонятельным и вкусовым). В организме больного образуется расширенная, патологическая система, в которую, по-видимому, входят эндокринная и иммунометаболическая системы. В пользу существования некоторого патологического образования, состоящего не только из мозга и кожи, свидетельствует тот факт, что среди дифференциальных признаков у пациентов с заболеванием «Психозуд» присутствуют признаки, указывающие на пережитые стрессы, нарушения в работе иммуннометаболической и эндокринной систем организма, а также наличие у кожи функции иммунитета. Если принять предлагаемую гипотезу, то зуд можно рассматривать как «обратный» (болезненный) эффект – т.е. как тактильные галлюцинации. Еще один аргумент в поддержку предлагаемой гипотезы – это появление галлюцинаций при отравлениях.

3.2. О создании модели

Итак, мы хотим создать модель патологической системы организма «Психозуд» в виде многоагентной системы. Ее интегративное свойство – образование псевдостатического объекта, т.е. поддержание равновесного состояния (например, гомеостаз), что характерно для природных систем. Для создания многоагентной модели патологической системы «Психозуд», прежде всего, следует определить, какие элементы в нее входят. Внутри псевдостатического объекта происходит взаимодействие элементов, которое порождает динамику и эмерджентное поведение элементов и самой системы. Поэтому следующим шагом является выяснение того, каким образом элементы системы взаимодействуют друг с другом, т.е. установление системообразующих связей (в данном случае связи могут осуществляться, например, с помощью гормонов). А затем необходимо создать правила динамического поведения элементов в системе. Поведение кожи, являющейся элементом системы, – функциональный зуд. Поведение мозга – психическое расстройство. Подчеркнем, что поведение элементов системы отличается от поведения независимых элементов. В рассматриваемой задаче это означает, что психическое расстройство больного с синдромом «Психозуд» иное, чем у больного с классическим психическим заболеванием: шизофрения, депрессия или истерия, по-видимому, в этих случаях отличаются друг от друга.

4. МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ СИСТЕМЫ «ПСИХОЗУД» В ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЕ ТИПА ДСМ

4.1. Использование ДСМ-метода для интеллектуального анализа данных о заболевании

Данными для интеллектуального анализа, реализуемого ДСМ-методом при создании модели патологической системы «Психозуд», являются описания пациентов с помощью качественных признаков, т.е. исследование начинается с анализа поведения всего организма в зависимости от различных факторов и установления элементов системы.

ДСМ-метод автоматического порождения гипотез имеет обширную библиографию (см., например, [18, 19]). Здесь скажем только, что в результате интеллектуального анализа данных с помощью ДСМ-метода получаются некоторые закономерности, связывающие фрагменты описаний объектов с наличием или отсутствием некоторого эффекта. В рассматриваемой задаче объекты – это описания пациентов, эффект – например, функциональный зуд. Эти зависимости могут быть использованы для диагностики заболевания. Кроме того, они могут рассматриваться как информация для определения элементов системы «Психозуд», системных связей, факторов внешней среды, генетических факторов, которые повлияли на возникновение заболевания. На основе этих зависимостей экспертами создаются правила динамического поведения элементов системы. Вообще, интеллектуальная система, реализующая ДСМ-метод, является человеко-машинной системой, т.е. работающей в полуавтоматическом режиме: подготовка данных, их изменение, пополнение, интерпретация результатов, снятие противоречий осуществляются совместно с экспертами.

4.2. Использование интеллектуальной системы типа ДСМ как инструмента моделирования

Для чего нужно моделирование? В первую очередь, для того, чтобы исследовать поведение реального объекта в некоторых условиях, которые трудно, нецелесообразно или невозможно создать. Моделирование реального объекта – это одновременно его исследование, поскольку практически никогда не удастся сразу получить хорошую модель. Моделирование – итерационный процесс.

Интеллектуальная система, реализующая ДСМ-метод (ИС-ДСМ), является средством создания теории предметной области [20]. В ИС-ДСМ можно добавлять и удалять объекты из обучающей выборки и, тем самым, изменять модель. Можно изменять непосредственно теорию (гипотезы) и проверять, правильно ли она работает.

ЗАКЛЮЧЕНИЕ

Создание моделей для исследования систем находится в настоящее время на переднем крае научных и технологических разработок в области искусственного интеллекта. Важнейшее средство для этого – многоагентный подход. Самым сложным этапом многоагентного моделирования является формулировка

правил динамического поведения агентов, т.е. здесь возникает та же проблема, что и раньше – проблема приобретения знаний существует в искусственном интеллекте еще со времен создания экспертных систем (ЭС) [21]. Интеллектуальный анализ данных, машинное открытие закономерностей, вообще машинное обучение помогали в решении этой проблемы для ЭС. Как развивать анализ данных для определения правил динамического поведения агентов? Этот вопрос пока не решен.

СПИСОК ЛИТЕРАТУРЫ

1. Романов Д. В., Львов А. Н. Психопатологический зуд в дерматологической практике // Доктор.Ру. – 2010. – № 4 (55). – С. 60-63.
2. Люгер Д. Искусственный интеллект: стратегии и методы решения сложных проблем. – М. : Издательский дом «Вильямс», 2003. – 864 с.
3. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – М. : Издательский дом «Вильямс», 2006. – 1408 с.
4. Johnson S. Emergence: The connected lives of ants, brains, cities, and software. - Scribner, 2001.
5. Финн В.К. Об интеллектуальном анализе данных // Новости искусственного интеллекта. – 2004. – № 3. - С. 3-18.
6. Финн В.К. Об определении эмпирических закономерностей посредством ДСМ-метода автоматического порождения гипотез // Искусственный интеллект и принятие решений. – 2010. - № 4. – С. 41-48.
7. Дюк В., Самойленко А. Data Mining: учебный курс. - СПб. : Питер, 2001. – 368 с.
8. Жилин Д.М. Теория систем. : Опыт построения курса. - М. : КомКнига, 2006. – 184 с.
9. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. - М. : Высшая школа, 1989. – 361 с.
10. Анохин П.К. Кибернетика функциональных систем. - М. : Медицина, 1998. – 397 с.
11. Флейшман Б.С. Системология, системотехника и инженерная экология // Кибернетика и ноосфера. - М.: Наука, 1986. – С. 97-100.
12. Агошкова Е.Б. Категория «система» в современном мышлении // Вопросы философии. - 2009. – № 4. - С. 57-71.
13. Бешенков С.А., Ракитина Е.А. Моделирование и формализация: методическое пособие. - М. : Лаборатория Базовых Знаний, 2002. – 336 с.
14. Уинстон П. Искусственный интеллект / под ред. Д.А. Поспелова. - М. : Мир, 1980. – 520 с.
15. Гаазе-Рапопорт М.Г., Поспелов Д.А. От амебы до робота: модели поведения. - М. : URSS, 2011. (2-е издание). – 296 с.
16. Эшби У. Росс. Введение в кибернетику / под ред. В.А. Успенского. - М. : КомКнига, 2005. (2-е издание). – 432 с.

17. Джарратано Д., Райли Г. Экспертные системы: принципы разработки и программирование. - М.: Изд. дом «Вильямс», 2007. – 1152 с.
18. ДСМ-метод автоматического порождения гипотез. Логические и эпистемологические основания / сост. О.М. Аншаков, Е.Ф. Фабрикантова; под общей ред. О.М. Аншакова. - М.: Книжный дом «Либроком», 2009. - 433 с.
19. Автоматическое порождение гипотез в интеллектуальных системах / сост. Е.С. Панкратова, В.К. Финн; под общ. ред. В.К. Финна. - М.: Книжный Дом «Либроком», 2009.- 528 с.
20. Фабрикантова Е.Ф. Описание моделей метаболизма веществ в виде квазиаксиоматических теорий // Новости искусственного интеллекта. Специальный выпуск, посвященный 60-летию В.К. Финна. - 1993. - С. 304-308.
21. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб. : Изд. дом «Питер», 2001. – 384 с.

Материал поступил в редакцию 07.08.12.

Сведения об авторе

ФАБРИКАНТОВА Елена Федоровна — кандидат технических наук, старший научный сотрудник ВИНИТИ РАН, Москва.
E-mail: fabr@viniti.ru.

К проблеме быстрого анализа заголовков при адресации пакетов данных в компьютерных сетях

Обсуждаются возможности использования технологий Программно-Конфигурируемых Сетей (ПКС-технологий) в оптимизации управления сетевым трафиком. Рассматриваются некоторые варианты сокращения перебора при анализе встречаемости заголовков пакетов данных в таблицах коммутации. Сформулированы условия и предложены алгоритмы, позволяющие оптимизировать процедуры проверки встречаемости входных заголовков в множествах строк таблиц коммутации больших размеров. Показаны возможности дополнительного ускорения обработки пакетов данных в компьютерных сетях с использованием ПКС-технологий.

Ключевые слова: программно-конфигурируемые сети, анализ заголовков пакетов данных в компьютерных сетях, математические методы анализа данных.

В современных компьютерных сетях одной из типовых процедур при адресации пакетов в коммутаторах оказывается поиск совпадения заголовка текущего (адресуемого) пакета со всеми элементами перечня заголовков, сопоставленных в таблице коммутации каждому выходному порту (см., например, [1] и др.). При, казалось бы, элементарном характере этой процедуры – проверке совпадения входной строки булевских переменных с одной или несколькими (при адресации обрабатываемого пакета сразу на несколько выходных портов) строками булевских переменных (в таблице коммутации) в ситуации, когда характерные размеры подобных таблиц – тысячи (а иногда и десятки тысяч) строк, а требования к скорости обработки каждого отдельного сообщения (например, в каналах с гигабитными показателями пропускной способности) являются крайне жесткими, любая возможность упростить процесс «опознания» требуемого для правильной адресации выходного порта (в частности – за счет сокращения объема и снижения вычислительной сложности выполняемых при этом операций) может дать дополнительные ресурсы для оптимизации обсуждаемой процедуры адресации.

Определенные надежды на успех в решении рассматриваемой задачи связывают с использованием возможностей так называемых Software-Defined Networks (SDN¹) [2- 6] и др. Принципиально важными здесь представляются возможности вынести на внешний (по отношению к конкретному коммутатору) контроллер хотя бы часть вычислений, требующих исполнения в процессе адресации. (Достаточно

гибкий доступ к необходимым вычислительным ресурсам в части как их объемов, так и производительности – вот основное преимущество технологий и решений этого типа).

В рамках SDN\ПКС-направления одним из наиболее популярных подходов к решению представленной задачи адресации заголовков в коммутационных таблицах является техника обобщений и развития так называемого геометрического подхода к классификации пакетов сообщений, предложенная Lakshman и Stiliadis [7-9], где формируемые правила классификации определяют соответствующие подпространства в исследуемом пространстве заголовков определенной размерности.

В основе предлагаемого нами подхода к решению обсуждаемой задачи лежит следующий набор простых соображений неформального характера. Если h_1 и h_2 – пара заголовков, адресуемых к одному и тому же выходному порту таблицы коммутации, а h_{12} – их общая часть, то вложимость подстроки h_{12} во вновь поступивший на вход процедуры адресации заголовок h_0 позволяет сузить объем необходимого в данном случае перебора вариантов со всей таблицы коммутации до ее фрагмента, содержащего все те строки, в которые входит подстрока h_{12} . Например, если h_{12} – максимальная общая часть h_1 и h_2 (и только их - !), то h_0 следует сравнивать лишь с h_1 и h_2 . Разумеется, учитывая при этом, что заранее, т. е. до запуска «рабочего» процесса адресации заголовков, следует также провести некоторые «подготовительные» процедуры, результатом которых и будет соотнесение каждого из «общих» фрагментов, например, h_{12} , – с релевантными ему строками в соответствующей таблице коммутации – например, соответственно с h_1 и h_2 .

¹ В русскоязычной литературе их обычно называют Программно-Конфигурируемыми Сетями (ПКС)

Возможности однократного проведения подобных «подготовительных» процедур для последующего многократного использования их результатов в процессе адресации пакетов по их заголовкам, усиленные как возможностями проведения предлагаемых вычислений на внешнем контроллере необходимой мощности, так и использованием аппаратно-схемных решений (например, на базе ТСАМ-технологий и др.) для проверки вложимости заранее выделенных «общих фрагментов» в конкретные заголовки адресуемых пакетов, открывают определенные позитивные перспективы при поиске высокопроизводительных решений для обсуждаемой задачи адресации.

При таком подходе задача распознавания адресуемого заголовка в таблице коммутации как задача своего рода дифференциальной диагностики заголовков пакетов по заданным выходным портам может быть переформулирована как задача выделения «разумного» множества соответствующих «дифференциальных признаков» (например, повторяющихся фрагментов заголовков, «отвечающих» за адресацию пакета на заданный выходной порт), на базе которого процедуру собственно распознавания адресуемого пакета в таблице коммутации можно будет «сузить» до просмотра лишь части этой таблицы, определяемой релевантными заголовку адресуемого пакета «дифференциальными признаками».

Таким образом, размерность (многократно) решаемой задачи распознавания (на всей таблице коммутации) может быть понижена за счет ее переформулирования в виде комбинации (однократно решаемой при формировании множества «дифференциальных признаков») задачи классификации и (также многократно решаемой) задачи собственно распознавания, но уже лишь на (релевантном «сработавшим» в анализируемом случае «дифференциальным признакам») фрагменте исходной таблицы коммутации.

При этом различные конкретные варианты балансировки «нагрузки» на объем (и соответственно, вычислительную сложность) ресурсов, затрачиваемых, с одной стороны,

на порождение подходящего варианта множества «дифференцирующих признаков» и, с другой стороны, на решение задачи распознавания, «усеченной» до просмотра лишь того фрагмента исходной таблицы коммутации, который соответствует выбранному множеству «дифференциальных признаков», позволяют сформировать соответствующее семейство конкретных алгоритмов адресации, каждый из которых характеризуется определенным уровнем распределения требуемых вычислительных ресурсов между порождением необходимого множества «дифференциальных признаков» и собственно поиском искомого строки в ограниченном (структурой выбранных классификационных признаков) фрагменте исходной таблицы коммутации. Некоторые характеристические особенности таких семейств алгоритмов и будут предметом нашего дальнейшего обсуждения.

Пусть задана коммутационная таблица **КТ**, которая образована n заголовками $h_1, h_2, \dots, h_n$ пересылаемых пакетов, адресуемых в свою очередь на N выходных портов $P_1, P_2, \dots, P_N$.

Будем говорить, что таблица **КТ** представлена в стандартной форме, если каждой ее строке сопоставлен в точности один выходной порт. (При этом в случае адресации конкретного пакета h_0 сразу на несколько выходных портов это выражается следующим образом: соответствующая строка-заголовок встречается³ в **КТ** ровно столько раз, сколько необходимо для перечисления всех необходимых в ее случае выходных портов). В свою очередь, будем говорить, что таблица **КТ** представлена в сжатой форме, если каждый заголовок представлен в ней соответствующей строкою ровно один раз, и при этом каждой такой строке сопоставлено все множество выходных портов, на которые должен быть адресован озаглавленный этой строкою пакет. Наконец, будем говорить, что таблица **КТ** представлена в форме, удобной для классификации, если она - в стандартной форме и разделена на N блоков, строки-заголовки каждого из которых адресуются на соответствующий выходной порт - $P_1, P_2, \dots, P_N$.

Будем называть подстроку $h_{i...j}$ ключом для (адресации) заголовков $h_i, \dots, h_j$ в таблице коммутации **КТ**, если $h_{i...j}$ вкладывается одновременно в заголовки $h_i, \dots, h_j$. Будем называть $h_{i...j}$ точным ключом для заголовков $h_i, \dots, h_j$, если $h_{i...j}$ - максимальная по вложению подстрока, которая одновременно вкладывается в таблицу коммутации **КТ** в эти и только в эти заголовки (в - $h_i, \dots, h_j$). Таким образом, точный ключ - это упорядоченная пара $\langle h_{i...j}, \{h_i, \dots, h_j\} \rangle$, где первая компонента пары - $h_{i...j}$ - максимальная по вложению общая подстрока для (представленного второй компонентой этой пары) множества $\{h_i, \dots, h_j\}$ заголовков из таблицы **КТ**, при этом вторая компонента рассматриваемой пары перечисляет все те и только те строки таблицы **КТ**, в которые входит подстрока $h_{i...j}$ (первый элемент рассматриваемой упорядоченной пары).

Пусть каждый из заголовков таблицы **КТ** представляет собой последовательность из n булевских символов - 0 или 1. Рассмотрим множество $U = \{a_{10}, a_{11}, a_{20}, a_{21}, \dots, a_{n0}, a_{n1}\}$, которое мы будем называть алфавитом, а его элементы - образующими. Сопоставим также каждой из строк h_i таблицы **КТ** соответствующую строку H_i , сформированную по следующему принципу: в качестве j -й компоненты строки H_i выбираются:

- 1) a_{j0} , если в заголовке h_i на позиции номер j стоит 0;
- 2) a_{j1} , если в заголовке h_i на позиции номер j стоит 1;
- 3) пара образующих a_{j0}, a_{j1} , если в заголовке h_i на позиции номер j стоит X^4 .

³ Причем, необязательно в рамках «односвязного» (последовательно перечисляющего все нужные выходные порты) фрагмента таблицы **КТ**.

⁴ Т.е. на позиции номер j в заголовке h_i может находиться любое из двух значений 0 или 1 (но только одно из них).

² Т.е. - «небольшого», обеспечивающего «полноту» процедуры адресации, ...

С полученной таким образом таблицей $КТ_H$ мы и будем далее работать.

Пусть зафиксировано множество заголовков $H_{1,r} = \{H_1, H_2, \dots, H_r\}$ и некоторое множество точных ключей $K_{i,j} = \{K_i, K_{i+1}, \dots, K_j\}$, где каждый ключ в этом множестве есть соответствующая (см. выше) упорядоченная пара вида $\langle \text{«максимальное» множество «общих» образующих из } U; \text{ множество «родителей» этого ключа - тех и только тех заголовков из } КТ, \text{ в которые вкладывается соответствующее} - \text{см. первую компоненту} - \text{«максимальное» множество «общих» образующих} \rangle$. Будем называть множество ключей $K_{i,j}$ точным относительно заданного множества заголовков $H_{1,r}$ тогда и только тогда, когда каждый из входящих в $H_{1,r}$ заголовков входит одновременно в множество $\{H_1, H_2, \dots, H_{s(l)}\}$ «родителей» каждого из точных ключей K_i из множества $K_{i,j}$. Будем называть множество ключей $K_{i,j}$ неизбыточным (относительно заданного множества заголовков $H_{1,r}$), если в общей части всех входящих в него множеств «родителей» нет ни одного элемента (заголовка из исходной таблицы $КТ$), не входящего в $H_{1,r}$. Наконец, будем называть множество точных ключей $K_{i,j}$ несократимым относительно $H_{1,r}$, если оно является неизбыточным и при удалении из $K_{i,j}$ любого из входящих в него ключей это свойство (неизбыточность) оказывается нарушенным.

Говоря неформально, точная (т.е. полная, неизбыточная и несократимая) система точных ключей (ТТ-система ключей) определяет подборку таких общих подзаголовков для множества заголовков $H_{1,r}$ из таблицы $КТ$, которых (и только которых) достаточно для идентификации в таблице $КТ$ всех элементов заданного множества $H_{1,r}$. В предложенном контексте представляется целесообразным для решения исходной задачи адресации искать ТТ-системы ключей с теми или иными экстремальными характеристиками, например, ТТ-системы, порождение которых требует вычислений минимальной сложности.

Обсуждая возможности порождения ТТ-систем ключей с заданными характеристиками, следует обратить внимание на следующие соображения. Пусть заданы множество образующих (алфавит) $U = \{a_{10}, a_{11}, a_{20}, a_{21}, \dots, a_{n0}, a_{n1}\}$ и построенное на его основе множество заголовков $H_{1,r} = \{H_1, H_2, \dots, H_r\}$, а $K(H_{1,r})$ – множество всех точных ключей для множества заголовков $H_{1,r}$. Рассмотрим следующие переборные задачи:

Задача 1. Точный ключ размера ровно k .

Дано: множества U и $H_{1,r}$.

Найти: существует ли в множестве $K(H_{1,r})$ точный ключ K_0 , сформированный ровно k различными образующими из множества U ?

Задача 2. Точный ключ, порожденный ровно k «родителями».

Дано: множества U и $H_{1,r}$.

Найти: существует ли в множестве $K(H_{1,r})$ точный ключ K_0 , сформированный ровно k «родителями» из множества $H_{1,r}$?

Задача 3. Число точных ключей.

Дано: множества U и $H_{1,r}$.

Найти: сколько элементов в множестве $K(H_{1,r})$?

Можно показать (см., например, [10] и др.), что имеют место следующие три утверждения:

Утверждение 1.

Задача Точный ключ размера ровно k принадлежит к классу NP-полных (см., например, [11] и др.) задач NPC.

Утверждение 2.

Задача Точный ключ, порожденный ровно k «родителями» принадлежит к классу NP-полных задач NPC.

Утверждение 3.

Задача Число точных ключей принадлежит классу #PC перечислительно полных задач.

Таким образом, попытки решить обсуждаемую задачу оптимизации адресации подходом, так сказать, «в лоб» (т.е. попытками либо найти «универсальный» ключ для всего $H_{1,r}$ прямым поиском, либо построить соответствующую систему ТТ-ключей их последовательным выбором из множества $K(H_{1,r})$ всех точных ключей для набора заголовков $H_{1,r}$) оказываются малопродуктивными, так как в общем случае ведут к экспоненциально сложным переборам.

Тем не менее, не сложно убедиться, что существенные сокращения перебора при адресации возможны. Пример таких возможностей демонстрирует даже следующий «наивный» алгоритм:

Алгоритм 1. (ПАДР⁵).

1. Выбираем два текущих элемента в множестве $H_{1,r} - H_{i1}$ и H_{i2} .
2. Вычисляем $h_{i1,i2}$ - максимальную общую часть заголовков H_{i1} и H_{i2} .
3. Дополняем множество $\{H_{i1}, H_{i2}\}$ всеми такими элементами исходного множества заголовков $H_{1,r}$, в которые также вкладывается $h_{i1,i2}$.
4. Удаляем из $H_{1,r}$ всех найденных таким путем «родителей» точного ключа $h_{i1,i2}$.
5. Повторяем процедуру, описанную шагами 1-4 на сокращенном представленном выше способом множестве $H_{1,r}$.
6. Если на предыдущем шаге (после удаления из рассмотрения очередного множества «родителей» очередного точного ключа) в рассмотрении остается менее двух заголовков из исходного множества $H_{1,r}$, то – СТОП.

Можно показать, что в худшем случае (в частности, для определенного вида⁶ множеств примеров – строк текущего состояния таблицы коммутации $КТ$) алгоритм ПАДР не дает каких-либо существенных преимуществ по сравнению с полным перебором строк таблицы коммутации $КТ$. Тем не менее, как несложно убедиться, при массовом применении (на

⁵ ПАДР - Первый Алгоритм Деревни Румянцево, названный так по месту его формирования: Бизнес-Центр «Румянцево», Московская обл., Ленинский р-н, деревня Румянцево.

⁶ В частности, можно подобрать такую таблицу коммутации, в которой строки расставлены (по возрастанию) в лексикографическом порядке и имеют такой вид, что для каждой последовательной (см. описание алгоритма ПАДР) пары строк их общая часть вкладывается в (специально подобранный) текущий заголовок h_0 , в то время как совпадает этот заголовок лишь с одним из элементов последней пары строк таблицы $КТ$.

плохо упорядоченных – см. сноску 7 - таблицах коммутации) этот «наивный» алгоритм дает сокращение перебора примерно в два раза.

Располагая множествами U^* и K^* , определим два отображения f и φ :

$$\forall A \in P(U^*) \quad f(A) = \{ h \in K^* \text{ таких, что для } \forall a \in A \text{ имеет место } a \in h \}$$

$$\forall B \in P(K^*) \quad \varphi(B) = \{ a \in U^* \text{ таких, что для } \forall h \in B \text{ имеет место } a \in h \}$$

(при этом $P(X)$ - множество всех подмножеств множества X (полагая $X \in \{U^*, K^*\}$). Несложно убедиться, что пара отображений $\langle f, \varphi \rangle$ представляет собою соответствие Галуа [12, 13], а их произведения $f(\varphi(B))$ и $\varphi(f(A))$ - соответствующие замыкания Галуа [12, 13]. Посредством $GC_{f,\varphi}(K^*)$ и $GC_{\varphi,f}(U^*)$ будем обозначать множества неподвижных точек соответствующих замыканий Галуа:

$$GC_{f,\varphi}(K^*) = \{ B \in P(K^*) \text{ таких, что } f(\varphi(B)) = B \},$$

$$GC_{\varphi,f}(U^*) = \{ A \in P(U^*) \text{ таких, что } \varphi(f(A)) = A \}.$$

Каждое из этих множеств можно рассматривать как частично упорядоченное в соответствии как со взаимным вложением соответствующих множеств заголовков (будем также называть их объектами), так и подмножеств образующих.

Неподвижными точками замыкания Галуа $\lfloor \rfloor_{KT,U^*}$ будем называть все такие $[h]_{KT,U^*}$, что:

$$[h]_{KT,U^*} = h.$$

Пусть $ДВ (GC(KT, U^*))$ – диаграмма взаимной вложимости замыканий Галуа $\lfloor \rfloor_{KT,U^*}$, построенных на объектах из KT (в свою очередь сформированных с помощью образующих из алфавита U^*). Легко видеть, что с точки зрения структуры частично упорядоченного множества $ДВ (GC(KT, U^*))$, алгоритм ПАДР оперирует с элементами верхней границы этой диаграммы. Альтернативой этому будет возможность оперировать с элементами нижней границы этого частично упорядоченного множества – замыканиями одноэлементных подмножеств алфавита U^* . Теперь мы обратимся к анализу свойств и возможностей этой системы ключей.

Определим (для приведения текущей таблицы коммутации KT к виду, удобному⁷ для порождения соответствующей диаграммы вложимости замыканий Галуа $\lfloor \rfloor_{KT,U^*}$) следующее

Правило нормализации (ПН).

Пусть задан заголовок $h = \{a_1, a_2, \dots, a_n\}$, в котором на позициях $i_1, i_2, \dots, i_s$ ($0 \leq s < n$) размещаются символы x (т.е. в каждой из этих позиций можно разместить как 0, так и 1). Множество $h^* = \{h, h^{i_0}, h^{i_1}, \dots, h^{i_s}, h^{i_1}\}$ строится так, что для каждого значения i из имеющегося множества $\{i_1, i_2, \dots, i_s\}$

⁷ «Удобство» в данном случае означает возможность отразить в соответствующей диаграмме вложимости все такие заголовки h , которые были бы сформированы лишь с помощью значений 0 и 1, а также порождались бы из имеющихся в KT строк с третьим значением x его заменой на 0 или 1.

номеров позиций символа x в заголовке h каждая пара h^{i_0}, h^{i_1} порождается из исходного заголовка h заменой x в соответствующей позиции один раз на 0, а другой – на 1 (см. таблицу).

Результатом применения **Правила ПН** к заданной таблице коммутации KT будет расширенная соответствующим образом таблица $KT^* = \{h_1, h_2, \dots, h_M\}$.

При этом имеет место

Утверждение 4.

При заданном n - размерности анализируемых заголовков h , построение (применением **Правила ПН**) по исходной таблице $KT = \{h_1, h_2, \dots, h_m\}$ расширенной таблицы коммутации $KT^* = \{h_1, h_2, \dots, h_M\}$ требует не более чем $O(n)$ операций. (Т.е. таблица KT^* строится по таблице KT полиномиально быстро).

Доказательство.

Расширенная таблица коммутации $KT^* = \{h_1, h_2, \dots, h_M\}$ порождается из исходной таблицы коммутации добавлением не более чем

$$M - n = 2s < 2n = O(n)$$

дополнительных новых строк (размера ровно n каждая).

Пусть:

$h = \{a_1, a_2, \dots, a_k\}$ - заголовок в расширенном алфавите $U^* = \{a_{01}, a_{11}, a_{02}, a_{12}, \dots, a_{0n}, a_{1n}\}$,

$KT = \{h_1, h_2, \dots, h_m\}$ - таблица коммутации (где h_i - ее строки, а $i \in \{1, 2, \dots, m\}$), переформулированная также в алфавит U^* ,

$KT^* = \{h_1, h_2, \dots, h_M\}$ - таблица коммутации KT , расширенная для каждого заголовка из множества $\{h_1, h_2, \dots, h_m\}$, содержащего соответствующую третьему значению x из множества $\{0, 1, x\}$ в позиции номер i пару $\{a_{0i}, a_{1i}\}$, дополнительными заголовками, порождаемыми по **Правилу ПН**,

$[b]_{KT,U^*}$ - определенное ранее замыкание Галуа (сформированное с учетом U^* как исходного алфавита и $KT = \{h_1, h_2, \dots, h_m\}$ как исходно заданного множества объектов - см. выше) для множества b ,

$GC(H, U^*)$ - множество всех замыканий Галуа (заданных по представленной схеме - см. выше), порожденных на заданном множестве объектов H , в свою очередь сформированных с помощью образующих из множества U^* .

Утверждение 5. (Теорема о представлении⁸).

Заголовок h вкладывается в одну из строк коммутационной таблицы KT (и при этом также содержится в множестве $GC(KT^*, U^*)$ замыканий Галуа, построенных на расширенной коммутационной таблице KT^*), тогда и только тогда, когда:

$$\bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*} = [h]_{KT^*, U^*} = [\{a_1, a_2, \dots, a_k\}]_{KT^*, U^*} = \left[\bigcup_{i=1}^k \{a_i\} \right]_{KT^*, U^*} = \{a_1, a_2, \dots, a_k\}. \quad (1)$$

⁸ Теорема о представлении Заголовка с помощью соответствующего множества ключей.

	α_1	α_2	...	α_i	...	α_j	...	α_{n-1}	α_n
	α_{01}, α_{11}	α_{02}, α_{12}	...	α_{0i}, α_{1i}	...	α_{0j}, α_{1j}	...	$\alpha_{0n-1}, \alpha_{1n-1}$	α_{0n}, α_{1n}
$\mathbf{h}$	a_1	a_2	...	X	...	X	...	a_{n-1}	a_n
$\mathbf{h}^{j0}$	a_1	a_2	...	0	...	X	...	a_{n-1}	a_n
$\mathbf{h}^{i1}$	a_1	a_2	...	1	...	X	...	a_{n-1}	a_n
...	...	...	...	...	...	...	...	...	...
$\mathbf{h}^{j0}$	a_1	a_2	...	X	...	0	...	a_{n-1}	a_n
$\mathbf{h}^{i1}$	a_1	a_2	...	X	...	1	...	a_{n-1}	a_n

Доказательство.

Прежде всего, представим условие доказываемого Утверждения в упрощенном виде:

$$\mathbf{h} \in \mathbf{GC}(KT^*, U^*) \iff^9 \bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*} = \mathbf{h}. \quad (1.1)$$

Далее учтем, что формула $\mathbf{h} \in \mathbf{GC}(KT^*, U^*)$ в соотношении (1.1) есть указание на то, что $\mathbf{h}$ – это неподвижная точка замыкания Галуа $[_]_{KT^*, U^*}$:

$$[\mathbf{h}]_{KT^*, U^*} = [\{a_1, a_2, \dots, a_k\}]_{KT^*, U^*} = \{a_1, a_2, \dots, a_k\}.$$

Кроме того, по определению замыкания Галуа $[_]_{KT^*, U^*}$ для всякого $\mathbf{h}_0$ имеет место:

$$\mathbf{h} \subseteq [\mathbf{h}]_{KT^*, U^*} \quad (1.2)$$

Таким образом, возможны всего два варианта для $[\mathbf{h}]_{KT^*, U^*}$:

$$\mathbf{h} = \{a_1, a_2, \dots, a_k\} = \bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*} \quad (1.2.1)$$

или

$$\mathbf{h} = \{a_1, a_2, \dots, a_k\} \subset \bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*}. \quad (1.2.2)$$

В случае (1.2.2) в множестве

$$\bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*}$$

помимо элементов $a_1, a_2, \dots, a_k$ содержится также некоторый элемент b_0 из U^* , такой, что:

$$b_0 \notin \mathbf{h} = \{a_1, a_2, \dots, a_k\} \quad (1.3)$$

при этом, по крайней мере одно из множеств $[\{a_j\}]_{KT^*, U^*}$ ($j = 1, 2, \dots, k$) также содержит этот элемент b_0 . Пусть это будет некоторое множество $[\{a_0\}]_{KT^*, U^*}$. Тогда, по определению замыкания Галуа $[_]_{KT^*, U^*}$, образующая a_0 входит в соответствующие объекты из KT^* только в паре с образующей b_0 . Другими словами: так как $[\{a_1, a_2, \dots, a_k\}]_{KT^*, U^*}$ есть множество всех образующих из U^* , которые одновременно входят в соответствующее подмножество

$$KT^*_{a_1, a_2, \dots, a_k} \subseteq KT^*, \quad (1.4)$$

то включая в себя a_0 , каждый объект $\mathbf{h}_0$ из этого подмножества $KT^*_{a_1, a_2, \dots, a_k}$ содержит также и b_0 . Т.е. в этом случае:

$$(\{a_1, a_2, \dots, a_k\} \cup \{b_0\}) \subseteq [\{a_1, a_2, \dots, a_k\}]_{KT^*, U^*}, \quad (1.5)$$

откуда следует:

$$\{a_1, a_2, \dots, a_k\} \subset [\{a_1, a_2, \dots, a_k\}]_{KT^*, U^*}. \quad (1.6)$$

Таким образом, в этом случае $\mathbf{h} = \{a_1, a_2, \dots, a_k\}$ не является неподвижной точкой замыкания Галуа $[_]_{KT^*, U^*}$:

$$\mathbf{h} \notin \mathbf{GC}(KT^*, U^*) \quad (1.7)$$

Теперь у нас имеются все необходимые инструменты, чтобы показать справедливость утверждения (1), переформулированного к виду (1.1).

(а) Необходимость.

Положим, что выполняется формула $\mathbf{h} \in \mathbf{GC}(KT^*, U^*)$, тогда выполняется лишь одно из соотношений (1.2.1) или (1.2.2). Пусть это будет (1.2.2), тогда, продвигаясь по цепочке рассуждений (1.3) – (1.7), мы приходим к противоречию с исходным допущением о том, что $\mathbf{h} \in \mathbf{GC}(KT^*, U^*)$. Таким образом, остается лишь вариант (1.2.1).

(б) Достаточность.

Положим, что выполнено соотношение

$$\bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*} = \mathbf{h}, \quad (1.8)$$

тогда, предположив, что

$$\mathbf{h} \notin \mathbf{GC}(KT^*, U^*), \quad (1.9)$$

и принимая во внимание, что выполнимость (1.9) сигнализирует о наличии среди элементов множества $a_1, a_2, \dots, a_k$ такого a_0 , что его замыкание $[\{a_0\}]_{KT^*, U^*}$ содержит также некоторый элемент b_0 , такой, что для него вновь выполняется условие (1.3):

$$b_0 \notin \mathbf{h} = \{a_1, a_2, \dots, a_k\}.$$

Далее по соображениям, учитывавшимся ранее в цепочке рассуждений (1.3) – (1.7), несложно прийти к соотношению

$$\left(\bigcup_{i=1}^k [\{a_i\}]_{KT^*, U^*} \right) \supset \mathbf{h},$$

которое противоречит исходно сделанному допущению (1.8).

⁹ Сокращенная запись для «формулы» *if and only if* (англ. - тогда и только тогда, когда).

В заключение обратим внимание на нетривиальный характер соотношения (1.8) (см. **Пример**), в рамках которого не требуется, чтобы *каждая* их входящих в **h** образующих соответствовала бы *неподвижной точке* замыкания Галуа $[_]_{KT^*, U^*}$. Требуется, чтобы объединение замыканий всех входящих в **h** образующих $a_1, a_2, \dots, a_k$ не выводило бы результат за пределы множества образующих $\{a_1, a_2, \dots, a_k\} = \mathbf{h}$.

Пример.

Положим $U^* = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, а $K = \{h_1, h_2, h_3, h_4\}$, где

$$\begin{aligned} h_1 &= \{a_1, a_2, a_4\}, \\ h_2 &= \{a_1, a_2, a_5\}, \\ h_3 &= \{a_1, a_3, a_6\}, \\ h_4 &= \{a_1, a_3, a_7\}. \end{aligned}$$

Несложно убедиться, что:

$$\begin{aligned} [\{a_1\}]_{K, U^*} &= \{a_1\}, \\ [\{a_2\}]_{K, U^*} &= \{a_1, a_2\} \supset \{a_2\}, \\ [\{a_3\}]_{K, U^*} &= \{a_1, a_3\} \supset \{a_3\}, \end{aligned}$$

тем не менее:

$$\begin{aligned} h_1 &= [\{a_1, a_2, a_4\}]_{K, U^*} = [\{a_4\}]_{K, U^*} = \{a_1, a_2, a_4\} \supset \{a_4\}, \\ h_2 &= [\{a_1, a_2, a_5\}]_{K, U^*} = [\{a_5\}]_{K, U^*} = \{a_1, a_2, a_5\} \supset \{a_5\}, \\ h_3 &= [\{a_1, a_3, a_6\}]_{K, U^*} = [\{a_6\}]_{K, U^*} = \{a_1, a_3, a_6\} \supset \{a_6\}, \\ h_4 &= [\{a_1, a_3, a_7\}]_{K, U^*} = [\{a_7\}]_{K, U^*} = \{a_1, a_3, a_7\} \supset \{a_7\}. \end{aligned}$$

Таким образом, для проверки присутствия заголовка **h** в коммутационной таблице **KT** следует убедиться, что **h** есть неподвижная точка соответствующего замыкания Галуа $[_]_{K, U^*}$, т.е. $[\mathbf{h}]_{K, U^*} = \{a_1, a_2, \dots, a_k\}$. При этом алгоритмически это можно сделать, проверив, например, выполнение условия:

$$\{a_1, a_2, \dots, a_k\} = \bigcup_{i=1}^k [\{a_i\}]_{K, U^*} \quad (2)$$

Замечание 1.

Соотношение (2) позволяет для диагностики встречаемости попадающих на вход заголовков **h** в коммутационной таблице **KT**:

а) один раз рассчитать необходимые значения для замыканий (в смысле текущего состояния коммутационной таблицы **KT**) всех элементов алфавита U^* ,

б) использовать эти расчеты многократно (в случае отсутствия изменений в таблице **KT**) для диагностики встречаемости в ней заголовков вновь приходящих на вход пакетов.

Замечание 2.

Соотношение (2) прямо не зависит от параметра m – числа строк в таблице коммутации **KT**, что позволяет проводить (после однократного вычисления замыканий для элементов множества U^*) диагностику встречаемости вновь попадающих на вход заголовков **h** за не зависящее от размеров таблицы **KT** время $T = \text{Const}(m)$.

При этом размер (число строк) таблицы **KT** существенен лишь при вычислении замыканий образующих из множества U^* с учетом объектов из

множества $KT = \{h_1, h_2, \dots, h_m\}$, в которые эти образующие входят.

Таким образом, рассматривая для каждого входного заголовка $\mathbf{h} = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ и таблицы KT^* соответствующее (эффективно порождаемое) подмножество $\{a_{i1}, a_{i2}, \dots, a_{in}\}$ образующих из алфавита U^* как основу для порождения соответствующей системы ключей

$$\begin{aligned} \mathcal{R}_{\mathbf{h}} &= \{[\{a_{i1}\}]_{KT^*, U^*}, [\{a_{i2}\}]_{KT^*, U^*}, \dots, [\{a_{in}\}]_{KT^*, U^*}\} \subseteq \\ &\subseteq \mathcal{R}_{KT^*} = \{[\{a_1\}]_{KT^*, U^*}, [\{a_2\}]_{KT^*, U^*}, \dots, [\{a_{2n}\}]_{KT^*, U^*}\}, \end{aligned}$$

можно убедиться, что (как следствие **Утверждения 5**) имеет место следующее

Утверждение 6.

Для каждого $\mathbf{h} = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ – заголовка длины n в алфавите $\{0, 1\}$ соответствующая система ключей

$$\mathcal{R}_{\mathbf{h}} = \{[\{a_{i1}\}]_{KT^*, U^*}, [\{a_{i2}\}]_{KT^*, U^*}, \dots, [\{a_{in}\}]_{KT^*, U^*}\}$$

является *полной, точной* и *несократимой*. При этом система ключей

$$\mathcal{R}_{KT^*} = \{[\{a_1\}]_{KT^*, U^*}, [\{a_2\}]_{KT^*, U^*}, \dots, [\{a_{2n}\}]_{KT^*, U^*}\}$$

является *полной, точной* и *несократимой* для множества всех строк в KT^* .

Доказательство.

Система ключей $\mathcal{R}_{KT^*}$ позволяет диагностировать те и только те заголовки, которые принадлежат таблице KT^* . Таким образом, обеспечены ее полнота и точность. Удаление из системы $\mathcal{R}_{KT^*}$ хотя бы одного элемента $[\{a_{i0}\}]_{KT^*, U^*}$, вообще говоря, нарушает ее полноту: например, можно подобрать такой $\mathbf{h}_0$, содержащий a_{i0} и KT^* , содержащую $\mathbf{h}_0$, что корректная диагностика принадлежности $\mathbf{h}_0$ к KT^* средствами системы ключей

$$\mathcal{R}_{\mathbf{h}_0} = \{[\{a_{i1}\}]_{KT^*, U^*}, [\{a_{i2}\}]_{KT^*, U^*}, \dots, [\{a_{i2n}\}]_{KT^*, U^*}\} \setminus [\{a_{i0}\}]_{KT^*, U^*}$$

будет невозможна. Таким образом, система ключей $\mathcal{R}_{KT^*}$ несократима.

Теперь, опираясь на возможности системы ключей $\mathcal{R}_{KT^*}$, сформулируем еще два быстрых алгоритма, позволяющих диагностировать встречаемость заголовка **h** вновь пришедшего на вход коммутатора пакета данных в коммутационной таблице KT^* .

Алгоритм 2. (ВАДР¹⁰).

1. Коммутационной таблице KT^* присоединим справа еще один столбец, для каждой строки вида **h** фиксирующий номер выходного порта i ($i = 1, 2, \dots, N$), на который пакет с заголовком **h** должен быть отправлен.

2. Отсортируем строки таблицы в порядке возрастания i – номеров (выходных портов) в ее последнем справа столбце.

¹⁰ ВАДР – Второй Алгоритм Деревни Румянцев.

3. Коммутационную таблицу KT^* разобьем на N подтаблиц $KT^*_1, KT^*_2, \dots, KT^*_N$ – по числу выходных портов рассматриваемого коммутатора.

4. Для каждой подтаблицы KT^*_i ($i = 1, 2, \dots, N$) восстановим алфавит $U^*_{KT^*_i}$ – соответствующее подмножество алфавита U^* .

5. Для каждого $i = 1, 2, \dots, N$:

5.1. Сформируем соответствующее множество ключей $\mathcal{R}_{KT^*_i}$.

5.2. Для каждого h – заголовка вновь приходящего на вход пакета данных – проверим выполнимость соотношения вида (2).

5.3. Выполнение этого соотношения означает встречаемость h в соответствующей таблице KT^*_i , что требует пересылки озаглавленного h пакета на выходной порт номер i рассматриваемого нами коммутатора.

5.4. Невыполнение соотношения вида (2) для h означает, что в таблице KT^*_i строка вида h отсутствует.

6. СТОП.

Алгоритм 3. (ТАДР¹¹).

1. Коммутационной таблице KT^* присоединим справа еще один столбец, для каждой строки вида h фиксирующий номера всех выходных портов i (где $i = 1, 2, \dots, N$), на которые пакет с заголовком h должен быть отправлен.

2. Сформируем соответствующее множество ключей $\mathcal{R}_{KT^*}$.

3. Для каждого h – заголовка вновь приходящего на вход пакета данных – проверим выполнимость соотношения вида (2).

4. Выполнение этого соотношения означает встречаемость h в таблице KT^* , что требует пересылки озаглавленного h пакета на выходные порты со всеми номерами i , перечисленными на пересечении строки h и последнего справа столбца расширенной (см. Шаг 1) таблицы KT^* .

5. Невыполнение соотношения вида (2) для h означает, что в таблице KT^* строка вида h отсутствует.

6. СТОП.

Замечание 3.

Оба представленных алгоритма – ВАДР и ТАДР – вполне заслуженно отнесены к классу быстрых алгоритмов:

- выполняемая на соответствующем Шаге процедура (для фиксированного состояния коммутационной таблицы KT – *однократного*!) порождения системы ключей $\mathcal{R}_{KT}$ требует для не более $2N$ циклов порождения замыкания $[\]_{KT, U^*}$ для каждого одноэлементного подмножества алфавита U^* (в свою очередь, процедуры линейной от размеров KT и U^* вычислительной сложности);

- а дальнейшие проверки выполнимости соотношения вида (2) оперируют лишь с не более чем $2N$

объектами из уже построенного GC -базиса, т.е. не зависят от текущего количества строк в таблице KT (что дает очевидные преимущества быстрой «диагностики» входного заголовка по отношению к коммутационным таблицам большого размера, при этом такие преимущества лишь увеличиваются при обработке растущего количества входных заголовков в текущем состоянии таблицы коммутации KT).

* * *

В порядке завершения предпринятого нами обсуждения обратимся к некоторым дополнительным возможностям, которые открывает в решении рассматриваемой задачи использование технологий Программно-Конфигурируемых Сетей (ПКС-технологий) и решений. Это возможности вынесения обеспечивающих вычислений (и прежде всего – порождения соответствующего GC -базиса) на специальный внешний контроллер. Управляя вычислительной мощностью этого контроллера, можно управлять также и скоростью обработки анализируемых заголовков пакетов. Дополнительный выигрыш здесь (как уже было показано выше) можно получить за счет многократного повторного использования уже построенного GC -базиса при «диагностике» текущих заголовков пакетов на входе коммутатора.

Не менее интересные возможности открывает дополнительное (в рамках ПКС-подхода) использование параллельных вычислений: параллельный расчет компонентов GC -базиса для каждой из образующих из алфавита U^* может быть совмещено, например, с параллельным анализом (в рамках реализации алгоритма ВАДР) соответствующих подтаблиц KT^*_i ($i = 1, 2, \dots, N$) анализируемой коммутационной таблицы KT^* по каждому из N выходных портов задействованного коммутатора. Аналогично, по уже сформированному GC -базису можно в параллельном режиме вести «диагностику» сразу нескольких входных заголовков.

Наконец, определенный потенциал оптимизации предлагаемой технологии «диагностики» входных заголовков представляется естественным связать и с использованием ТСАМ-решений (например, для проверки представимости текущего заголовка в уже построенном GC -базисе и др.). Разумеется, все эти возможности требуют отдельного изучения и подробного обсуждения – но уже за рамками настоящей работы.

СПИСОК ЛИТЕРАТУРЫ

1. Смелянский Р.Л. Компьютерные сети (в 2-х томах). - М: Издательский центр "Академия", 2011. (Том 1. Системы передачи данных. - 304 с., Том 2. Сети ЭВМ. - 240 с.)
2. Исследовательский проект 7-й Рамочной Программы Европейского Союза «FEDERICA». - URL: <http://dana.i2cat.net/the-end-of-federica-project/platform/>
3. Национальная Программа США «Global Environment for Network Innovations (GENI)». - URL: <http://www.geni.net>

¹¹ ТАДР – Третий Алгоритм Деревни Румянцево.

4. Исследовательский проект 7-й Рамочной Программы Европейского Союза «OFELIA». - URL: <http://www.fp7-ofelia.eu/>
5. Ассоциация «Open Networking Foundation». - URL: <https://www.opennetworking.org/>, <http://www.openflow.org>
6. Open Networking Laboratory\ Open Networking Research Center. – URL: <http://onrc.stanford.edu>
7. Lakshman T.V., Stiliadis D. High-speed policy based forwarding using efficient multi-dimensional range matching. - Tech. Rep. 113470-980202-02T, Lucent Technologies-Bell Labs., Jan. 1998; also in: Proceedings of ACM SIGCOMM'98/ - Vancouver, CA, September 1998.
8. Kumar V.P., Lakshman T.V., Stiliadis D. Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet. – In: IEEE Communications. - May 1998. - Vol. 36, No. 5. - P. 152-163.
9. McKeown N., Kazemian P., Varghese G. Header Space Analysis: Static Checking for Networks. – In: 9th USENIX Symposium on Networked Systems Design and Implementation. NSDI '12, April 2012, San Jose, CA.
10. Забежайло М.И. О некоторых оценках сложности вычислений в ДСМ-рассуждениях. (*в печати*).
11. Гэри М., Джонсон Д.С. Вычислительные машины и трудно-решаемые задачи. - М.: Мир. - 1982. - 416 с.
12. Гусакова С.М., Финн В.К. Сходства и правдоподобный вывод // Известия АН СССР. Сер. Техническая Кибернетика. -1987. - N 5. - С.42-63.
13. Кон П. Универсальная алгебра - М.: Мир, 1968. – 359 с.

Материал поступил в редакцию 27.09.12

Сведения об авторе

ЗАБЕЖАЙЛО Михаил Иванович – кандидат физико-математических наук, управляющий директор Центра прикладных исследований компьютерных сетей, Москва (Сколково)
E-mail: zmivan@gmail.com

В.Ф.Головин, В.В.Журавлев, М.В.Архипов, В.Е.Павловский, И.А.Орлов

Особенности математического моделирования многосуставного робота, взаимодействующего с упругой динамической средой

Рассматриваются вопросы моделирования динамики многозвенных манипуляторов в новых задачах роботизированной механотерапии на примере массажа и разработки конечностей пациента, а также синтез и моделирование их управления, анализ зависимости параметров двигателей в сочленениях от характеристик рабочей поверхности. Исследуется влияние внешних факторов на безопасность механотерапии. Описываются несколько комплексных моделей и финальная модель манипулятора РМ-01, его операционная среда построена и исследована с помощью программного комплекса «Универсальный механизм» (УМ), управление реализовано в пакете MATLAB Simulink.

Ключевые слова: моделирование движения, позиционно-силовое управление, манипулятор РМ-01, механотерапия, управление, «Универсальный механизм», MATLAB Simulink

ВВЕДЕНИЕ

Управление манипуляционными роботами, обеспечивающее сложные двигательные функции, необходимые в восстановительной медицине, а именно в массаже и проработке движений конечностей в суставах, рассмотрено в ряде работ (например [1-4]). В роботизированной механотерапии возможны две полярные тенденции создания манипуляционных роботов: разработка универсальных многосуставных роботов и узкоспециализированных простых одно-, двухстепенных роботов. Между этими классами имеется ряд роботов с разным количеством степеней подвижности, специализированных на определенные задачи. Необходимость разработки многосуставных роботов вызывается сложностью пространственных траекторий при массаже для различных участков тела и разнообразием применяемых приемов. Роботы могут быть оснащены различного уровня сложности системами позиционно-силового и биотехнического управления [5,6]. Это требует, вообще говоря, применения сложных развитых средств моделирования при отработке управления такими роботами [7,8].

В работе рассматриваются проблемы, возникающие в связи с реализацией метода позиционно-силового управления (ПСУ) с силовым обучением [5]. Суть метода состоит в том, что, в отличие от распределенного позиционного обучения, обученные точки содержат также информацию об усилиях и деформациях. В большой степени такой способ эффективен для сильно деформируемых биологических мягких тканей, на которых робот манипулирует, выполняя приемы массажа и движения конечностей в суставах [9,10].

Эффективность выполнения массажных приемов в постановке задачи ПСУ с силовым обучением опре-

деляется точностью обеспечения заданных усилий на траектории перемещения инструмента. Существует ряд причин, связанных не только с особенностями раздельного ПСУ с силовым обучением, но и с биологической природой мягких тканей пациента, снижающих точность обеспечения заданных усилий.

Прежде всего, несмотря на то, что методика проведения массажа предполагает неподвижность пациента, он не в состоянии осуществить задержку дыхания в течение длительного времени. При дыхании, в наибольшей степени, смещаются поверхности, а, следовательно, и обученные точки, в районе грудной клетки и живота пациента, в наименьшей степени - в области конечностей. Возникают также смещения обученных точек вдоль поверхности тела по другим причинам. Если такие смещения вызывают недопустимые отклонения усилия от заданного при позиционном отслеживании силовых обученных точек, то необходимо применять ПСУ с силовой коррекцией обученных точек.

Внешние факторы, осложняющие функционирование робота при механотерапии, таковы:

1. Наличие под упругой мягкой тканью (МТ) жесткой костной ткани.
2. Подход к поверхности мягкой ткани, под которой на неизвестном (и варьируемом для разных людей) расстоянии находится жесткая костная ткань.
3. Ограничения модели контактного взаимодействия, в которой предполагается равенство нормальных усилий при силовом обучении и позиционном слежении.
4. Отклонение интерполированной по обученным точкам траектории от той, на которой обеспечиваются заданные усилия.
5. Отклонение инструментальной оси от нормали к поверхности.

Рассмотрим далее основные задачи и модели роботизированной механотерапии.

Рука пациента может быть полностью пассивно расслабленной, напряженной за счет волевых усилий, напряженной за счет функциональных или органических повреждений.

Рассмотрим кинематическую схему двух связанных рук: пациента и робота. Кисть пациента зафиксирована в схвате робота (рис.1). Эта система представляет замкнутую кинематику. Руки пациента и робота в данном примере представляются ангулярными кинематическими моделями, хотя робот может иметь другую кинематику, например цилиндрическую в зависимости от задачи манипулирования (плечевые и кистевые суставы обозначены шаровыми шарнирами).

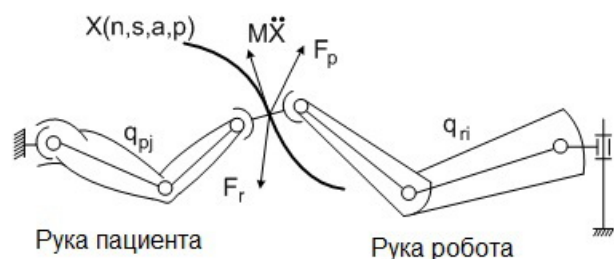


Рис.1. Схема взаимодействия руки робота с рукой пациента

В 12-мерном представлении вектор $\mathbf{X}(t)$ имеет вид:

$$\mathbf{X}(t) = (\mathbf{n}, \mathbf{s}, \mathbf{a}, \mathbf{p}),$$

где $\mathbf{p}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ – вектор положения, $\mathbf{n}, \mathbf{s}, \mathbf{a}$ – векторы, характеризующие ориентацию.

В 12-мерном представлении вектор $\mathbf{X}(t)$ имеет вид:

$$\mathbf{X}(t) = (\mathbf{n}, \mathbf{s}, \mathbf{a}, \mathbf{p}),$$

где $\mathbf{p}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ – вектор положения, $\mathbf{n}, \mathbf{s}, \mathbf{a}$ – векторы, характеризующие ориентацию.

На рис.1 $\mathbf{q}_p$ и $\mathbf{q}_r$ – суставные координаты пациента и робота. Таким образом,

$$\mathbf{X}(t) = f_p(\mathbf{q}_p) = f_r(\mathbf{q}_r),$$

где f_p и f_r – функции, определяющие прямую кинематическую задачу о положениях рук пациента и робота.

Для определения суставных координат робота, обрабатываемых приводами, обратная задача формулируется следующим образом:

$$\mathbf{q}_p \rightarrow \mathbf{X} \rightarrow \mathbf{q}_r \text{ или } \mathbf{q}_r = f_r^{-1}(f_p(\mathbf{q}_p)).$$

При известных кинематических моделях рук пациента и робота алгоритм обратной задачи кинематики может использоваться, например, для виртуального моделирования.

Для двух соединенных рук робота и пациента система дифференциальных уравнений динамики в форме Лагранжа записана ниже (1). Третье уравнение представляет движение по траектории $\mathbf{X}(t)$ кисти пациента, закрепленной в схвате робота.

$$\begin{cases} \mathbf{A}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{B}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) = \mathbf{F}_{rd} - \mathbf{J}_r^T(\mathbf{q}_r)\mathbf{F}_p, \\ \mathbf{A}_p(\mathbf{q}_p)\ddot{\mathbf{q}}_p + \mathbf{B}_p(\mathbf{q}_p, \dot{\mathbf{q}}_p) = \mathbf{F}_{pm} - \mathbf{J}_p^T(\mathbf{q}_p)\mathbf{F}_r, \\ \mathbf{M}\ddot{\mathbf{X}} = \mathbf{F}_r + \mathbf{F}_p. \end{cases} \quad (1)$$

$\mathbf{A}_r(\mathbf{q}_r)$ и $\mathbf{A}_p(\mathbf{q}_p)$ – матрицы, учитывающие инерционные свойства робота и конечности пациента – соответственно, $\mathbf{B}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)$ – вектор, учитывающий гравитационные, центробежные и кориолисовы силы манипулятора, $\mathbf{B}_p(\mathbf{q}_p, \dot{\mathbf{q}}_p)$ – вектор, учитывающий упругие и вязкие силы мышц конечностей пациента, $\mathbf{M}$ – матрица, учитывающая инерционные свойства части конечности пациента, закрепленной в схвате робота, $\mathbf{J}_r(\mathbf{q}_r)$ – матрица Якоби манипулятора робота, $\mathbf{J}_p(\mathbf{q}_p)$ – матрица Якоби конечности пациента, $\mathbf{F}_{rd}$ – усилие, развиваемое приводами робота, $\mathbf{F}_{pt}$ – усилие, развиваемое мышцами конечности пациента, $\mathbf{F}_r$ – усилие, развиваемое роботом на схвате, $\mathbf{F}_p$ – усилие, развиваемое конечностью.

В первое уравнение системы (1) входит выражение для активной силы мышц конечности $\mathbf{F}_p$, в отличие от уравнения, описывающего манипуляции робота на пассивной мягкой ткани [9, 10]. В диагностической измерительной задаче ведущим может быть пациент, тогда рука робота разгружается или оказывает сопротивление.

В соответствии с кинематической и динамической моделями системы «рука робота – рука пациента» приведена структурная схема на рис.2.

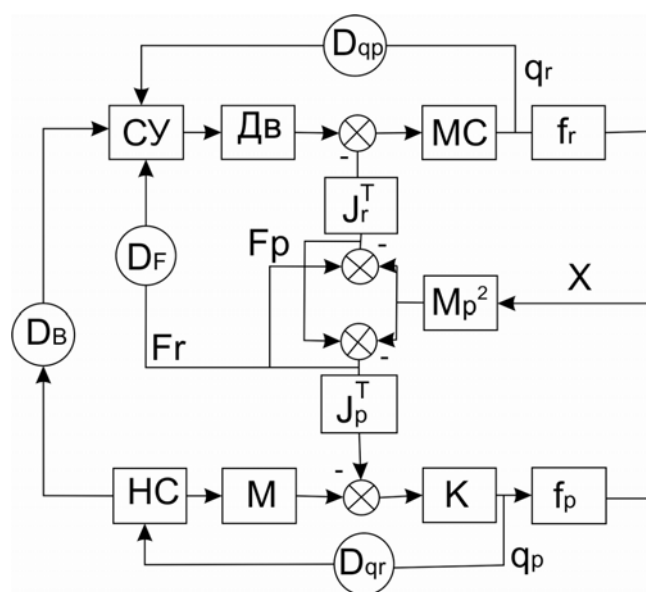


Рис.2. Структурная схема позиционно–силового управления
системой рука робота рука пациента

Верхняя часть схемы представляет робот с манипуляционной системой МС, двигателями приводов Дв, системой управления СУ, датчиком усилия D_F и датчиками положения и скоростей звеньев робота Dq_r . В нижней части схемы представлена конечность – рука пациента К, перемещаемая мышцами М, управляемыми нервной системой НС, воспринимающей сигналы рецепторов мышц Dq_p . Для управления роботом могут использоваться также биомедицинские сведения, например, через биопотенциалы при помощи датчика D_B .

Для данной структурной схемы могут быть реализованы три вида позиционно-силового управления: независимое, когда управления по силе и положению разделены во времени, согласованное, комбинированное.

В случае, когда для обучения и упражнения поврежденной руки пациента робот старается перемещать схват с захваченной кистью по заданной траектории $X_0(t)$, он должен уступать спазмированной напряженной руке пациента в направлении сил сопротивления. Рука робота должна быть упруго податливой. При упругом импедансном управлении [11] усилие, развиваемое роботом для восстановления заданной траектории, будет пропорционально ошибке (податливости):

$$F_r = K_s (X_0(t) - X(t)) + K_d (\dot{X}_0(t) - \dot{X}(t)),$$

где K_s и K_d – матрицы жесткости и скоростной податливости.

Если воспользоваться эталонной моделью объекта, описываемой векторным линейным дифференциальным уравнением второго порядка, получим уравнение:

$$\ddot{X}_0 + G_1 \dot{X}_0 + G_2 X_0 = 0,$$

где G_1 и G_2 – матрицы, обеспечивающие желаемое качество переходных процессов.

Закон управления по усилиям, развиваемым приводами, будет иметь вид:

$$F_{rd} = A_r(q_r) [\ddot{q}_{r0} + G_1(\dot{q}_{r0} - \dot{q}_r) + G_2(q_{r0} - q_r)] + B_r(q_r, \dot{q}_r) - J_r^T(q_r) F_r.$$

Для контроля усилий F_r на пространственной траектории шестикомпонентный силовой датчик размещается в запястье робота.

МОДЕЛЬ РОБОТА, ВЗАИМОДЕЙСТВУЮЩЕГО С ДИНАМИЧЕСКОЙ СРЕДОЙ

Динамической средой будем называть неопределенную, или подвижную, или сопротивляющуюся движениям робота среду. Особое внимание при этом обращается на невозможность при взаимодействии робота с неопределенной средой одновременно точно управлять и усилием и перемещением. Если среда известна и описывается в общем случае как динамическая с известной динамикой, то предлагается ряд

алгоритмов ПСУ. Робот, взаимодействующий с динамической средой, может быть представлен следующей моделью:

$$\begin{cases} H(q)\ddot{q} + h(q, \dot{q}) = \tau + J^T(q)F, \\ \tilde{M}(q)\ddot{q} + \tilde{L}(q, \dot{q}) = -J^T(q)F, \end{cases}$$

где $q, \dot{q}, \ddot{q} \in R^n$ – векторы обобщенных (суставных) координат положения, скорости, ускорения соответственно; $H(q)$ – матрица, характеризующая инерционные свойства манипулятора; $h(q, \dot{q}) \in R^n$ – вектор, характеризующий гравитационные, центробежные, кориолисовы силы в звеньях манипулятора; $\tilde{M}(q) \in R^{n \times n}$ – матрица, характеризующая инерционные свойства среды; $\tilde{L}(q) \in R^n$ – вектор, характеризующий вязкоупругие свойства среды и ее вес; $\tau \in R^n$ – вектор управляющих моментов, развиваемых приводами манипулятора; $F \in R^m$ – вектор сил, действующих на конечное звено (инструмент) манипулятора со стороны среды; $J(q) \in R^{m \times n}$ – матрица Якоби, связанная с преобразованием скоростей обобщенных координат $\dot{q}$ и декартовых координат конечного звена $\dot{X}$:

$$\dot{X} = J(q)\dot{q}.$$

Если $m = n$, то среда описывается уравнением

$$M(q)\ddot{q} + L(q, \dot{q}) = -S^T(q)F,$$

где $S(q) \in R^{n \times n}$ – матрица ранга n . В этом случае одним из законов динамического управления, основанном на решении обратной задачи динамики (или методе вычисляемых моментов), является закон управления с обратными связями по $q, \dot{q}, F$:

$$\tau = H(q)M^{-1}(q) \left[-L(q, \dot{q}) - S^T(q)(F_0 + \int_{t_0}^t Q(\mu(\omega))d\omega) \right] + h(q, \dot{q}) - J^T(q)F.$$

Здесь F_0 – заданное значение усилия, $\mu(t) = F(t) - F_0(t)$ – ошибка воспроизведения заданного усилия, $Q(\mu) \in R^n$ – непрерывная вектор-функция, определяемая уравнением $\dot{\mu}(t) = Q(\mu(t))$.

Вообще говоря, экспоненциальная и практическая устойчивость системы с приведенным законом управления доказана. Однако реализация такого закона управления затруднительна, поскольку требуются точные модели манипулятора и среды, особенно, если говорить о матрицах $H(q)$, $M(q)$ и $J(q)$.

МОДЕЛИ РЕЛЬЕФА ПОВЕРХНОСТИ МЯГКОЙ ТКАНИ

Существенной для деформирования биологических мягких тканей является ошибка $\xi(t)$, представляющая отклонение интерполированной роботом траекто-

рии от той идеальной, на которой обеспечивается точное заданное усилие. Здесь и далее считаем обучение робота традиционным поточечным. В обученных точках можно обеспечить $\xi(t) = 0$, а между обученными точками, особенно, если их немного, из-за неопределенности рельефа ошибка $\xi(t)$ может быть значительной. Рассмотрим ряд моделей процесса $\xi(t)$, полагая, что интерполяция по обученным точкам линейная и инструмент робота в каждой обученной точке ориентируется по нормали к поверхности МТ.

Модель 1. Рельеф поверхности МТ между соседними обученными точками представляет дугу окружности радиуса R , центральный угол этой дуги равен φ , см. рис. 3.

Пусть для соседних точек A_1 и A_2 дуга имеет наибольшее удаление от хорды A_1A_2 равное x_1 (рис.3). Смещение x_1 вызывает силовую ошибку:

$$\varepsilon_F = kx_1,$$

где k – коэффициент упругости среды под дугой A_1A_2 , $x_1 = 2R \sin^2(\varphi/4)$.

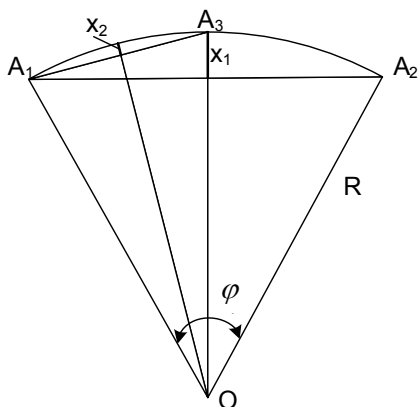


Рис.3. Схема интерполирования точек дуги

Добавляем обученную точку A_3 на середину дуги A_1A_2 . Наибольшее удаление дуги A_1A_3 от хорды A_1A_3 равно: $x_2 = 2R \sin^2(\varphi/8)$.

При разбиении дуги на n участков имеем: $x_n = 2R \sin^2(\varphi/4n)$. Число обученных точек N при разбиении на n участков увеличивается по формуле:

$$N = 2 + \sum_{i=1}^n 2^{i-1}.$$

Таким образом, в сравнении с интерполяцией по двум крайним точкам дуги, интерполяция по $n+1$ точке уменьшает силовую ошибку в

$$m \text{ раз, } m = \frac{\sin^2(\varphi/4n)}{\sin^2(\varphi/4)}.$$

При малых углах φ , когда $\sin \varphi \approx \varphi$, силовая ошибка уменьшается в n раз. Предлагаемая модель адекватна для систем ПСУ с силовым обучением, когда на участках между соседними обученными точками нет точек, в которых $F = F_0$.

Модель 2. Рельеф поверхности МТ между соседними обученными точками представляет синусоиду, нули которой совпадают с обученными точками.

$$\xi(t) = a \sin\left(\frac{2\pi}{T}t\right),$$

где T – удвоенное расстояние между соседними обученными точками.

При увеличении числа обученных точек амплитуда периодического колебания будет уменьшаться, а частота повышаться.

Модель 3. Рельеф поверхности МТ представляет дискретный случайный процесс с шагом квантования, равным расстоянию между обученными точками и значениями решетчатой функции, равными средним значениям процесса на шаге квантования.

Модель 4. Рельеф поверхности МТ представляет непрерывный стационарный случайный процесс с корреляционной функцией $R(\tau)$. При увеличении числа обученных точек будет уменьшаться вес низкочастотных составляющих, другими словами, увеличение числа обученных точек исполняет роль низкочастотной фильтрации. Модель адекватна при однородности рельефа поверхности МТ.

Алгоритм ПСУ с силовым обучением использовался в большинстве исследований с роботом РМ-01 [5]. В частности, для оценки ошибки интерполяции $\xi(t)$ были проведены экспериментальные исследования на рельефном макете с твердой поверхностью и на длинных мышцах спины человека.

В одном из опытов на макете спины были обучены вручную и автоматически 42 точки на расстоянии 10 мм друг от друга, с усилием 10Н с точностью $\pm 5\%$. Инструмент робота – твердый шарик диаметром 10 мм.

Если точки обучались вручную с малой скоростью, то ошибки отслеживания траектории, интерполированной по этим точкам, были незначительны. Для повышения скорости обучения необходимо увеличивать шаг перемещения, что вызывает большие ошибки. На рис.4 приведена ошибка $\xi(t)$ как разность между заданной траекторией (интерполированной по 42 точкам, обученным вручную с малой скоростью) и полученная интерполированием по пяти точкам, а также усилия, возникающие при прохождении по точкам, обученным с различными шагами.

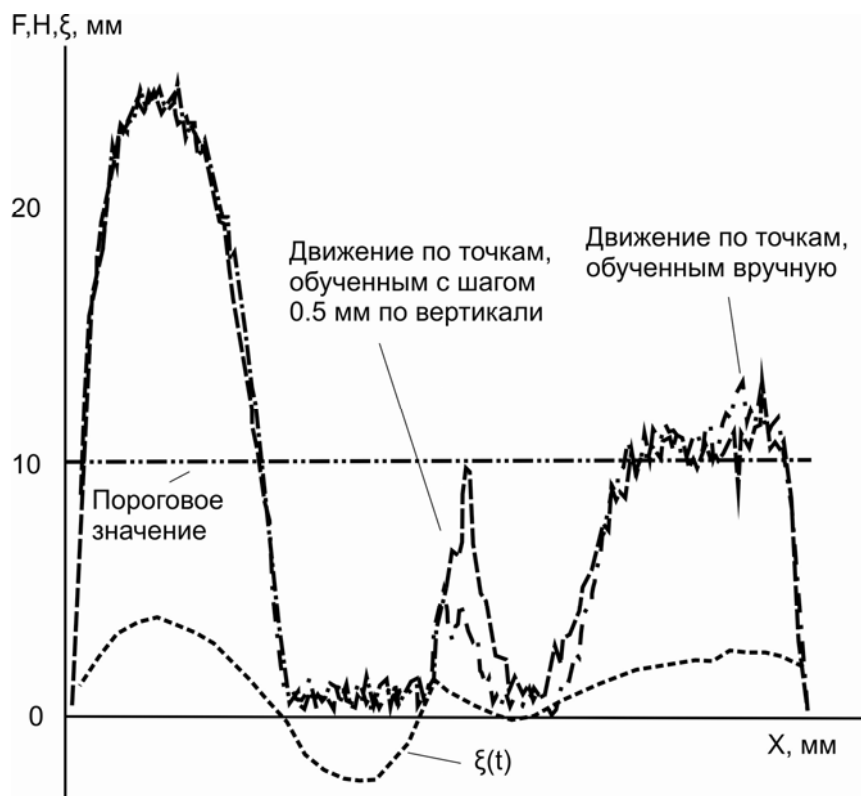


Рис. 4. Графики усилий и ошибки интерполяции при движении по 5 точкам

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ МНОГОСУСТАВНЫХ РОБОТОВ НА ОСНОВЕ ПРИНЦИПА НАИМЕНЬШЕГО ПРИНУЖДЕНИЯ ГАУССА

А.Ф. Верещагиным в МВТУ им. Н.Э. Баумана была разработана на основе принципа наименьшего принуждения Гаусса программа «Робот».

Принцип принуждения по Гауссу состоит в том, что истинные ускорения системы материальных точек, находящейся под действием активных сил и подчиненной идеальным связям, доставляют минимум принуждению по Гауссу [12].

Принцип Гаусса для задачи определения движения механической системы под действием заданных сил дает следующее решение: при заданной конфигурации и скорости механизма вычисляются ускорения при минимизации меры принуждения с учетом уравнений связей. Скорости и полученные ускорения интегрируются для определения новой конфигурации и новых скоростей в следующий момент времени.

Такой процесс определения траектории соответствует численному интегрированию уравнений движения на ЭВМ. Однако здесь можно миновать стадию вывода уравнений движения, извлекая всю необходимую информацию алгоритмическим путем из решения задачи минимизации.

Принцип наименьшего принуждения является одним из наиболее общих принципов. Из него, в частности, вытекают уравнения Лагранжа второго рода как реализация необходимого условия минимума меры принуждения. Однако достоинством использования принципа наименьшего принуждения является отказ от вывода уравнений движения в замкнутой форме и

определение ускорений системы путем решения вариационной задачи. Так как мера принуждения составляет квадратичную форму, то вариационная задача решается путем минимизации квадратичного функционала с учетом линейных ограничений.

В программе «Робот» исполнительный механизм задается в виде структурной схемы (рис. 5). На рис. 5 q_1, q_2 — обобщенные координаты робота, q_{p1}, q_{p2} — программные обобщенные координаты робота, M_{n1}, M_{n2} — моменты (усилия), развиваемые приводами.

Структурные схемы приводных устройств кодируются на входном языке из типовых функциональных блоков. Библиотека типовых блоков содержит различные линейные динамические блоки, задаваемые передаточными функциями, нелинейные блоки, логические и импульсные элементы, генераторы случайных процессов, формирователи критериев качества. Входной язык предусматривает задания на моделирование, многовариантный анализ и оптимизацию.

На рис. 6 приведена кинематическая схема четырехзвенного ангулярного робота с параметрами, близкими к роботу РМ-01. В табл. 1 и 2 приведены массоинерционные характеристики и параметры Денавита-Хартенберга для данного робота, соответственно.

Все приводы имеют последовательную коррекцию в виде ПД-регуляторов. Коэффициент $K_{\Pi} = 3000$ определяет жесткость привода, коэффициент $K_{\text{д}} = 300$ определяет степень демпфирования. Коэффициенты выбраны из условия аперiodичности процессов всех приводов.

На рис. 7 представлены аperiodические переходные процессы перемещения по высоте конечного звена манипулятора, слева для $m_4 = 1$, справа для $m_4 = 3$. Ось времени в секундах – по вертикали.

Таблица 1

	1	2	3	4
m	20	18	5	1
I	0,1	0,5	0,5	0,1
X_c	0	-0,25	-0,25	-0,005
Y_c	-0,05	0	0	0
Z_c	0	0	0	0
I_x	0,2	0,2	0,07	0,01
I_y	0,5	0,5	0,02	0,01
I_z	0,5	0,6	0,1	0,002
φ_d	1,5	5	1	0,4

Таблица 2

	θ	η	v	α
$-1 \rightarrow 0$	0	0	0	0
$0 \rightarrow 1$	1,57	0,1	0	1,57
$1 \rightarrow 2$	0	0	0,5	0
$2 \rightarrow 3$	0	0	0,5	0
$3 \rightarrow 4$	0	0	0,1	0

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ МНОГОСУСТАВНЫХ РОБОТОВ НА ОСНОВЕ УРАВНЕНИЙ ЛАГРАНЖА 2-ГО РОДА (ПРОГРАММНЫЙ ПАКЕТ MBTU-3.7)

Манипулятор представляется в виде блока, в котором учитывается его конфигурация, массо-инерционные характеристики и кинематика. На рис. 8 приведена модель четырехзвенного манипулятора (рис.6) с массо-инерционными характеристиками и кинематикой, определенными в табл. 1, 2.

На рис.9 приведены переходные процессы суставных координат, на рис.10 дан переходной процесс перемещения конечного звена: слева при $m_4 = 1$, справа переходной процесс перемещения конечного звена при $m_4 = 3$. Здесь $q_2 - q_4$ – суставные координаты, z – перемещение конечного звена

Следует отметить, что парадигма моделирования, реализованная в программе “MBTU-3.7” (как это, в частности, показывает рис.8), существенно напоминает подход, принятый в пакете MATLAB-Simulink, но мощность последнего и набор программных примитивов в нем значительно шире. Отметим также, что при работе в пакетах «Робот» и «MBTU-3.7» велик объем ручной работы при синтезе моделей, так как эти пакеты не имеют развитых средств автоматизации моделирования. Этих недостатков лишен программный пакет «Универсальный механизм» [7,13]. Покажем особенности его использования.

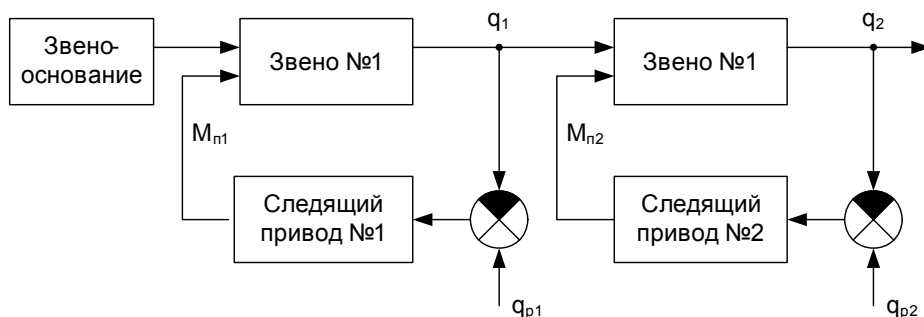


Рис.5. Структурная схема многосуставного исполнительного механизма

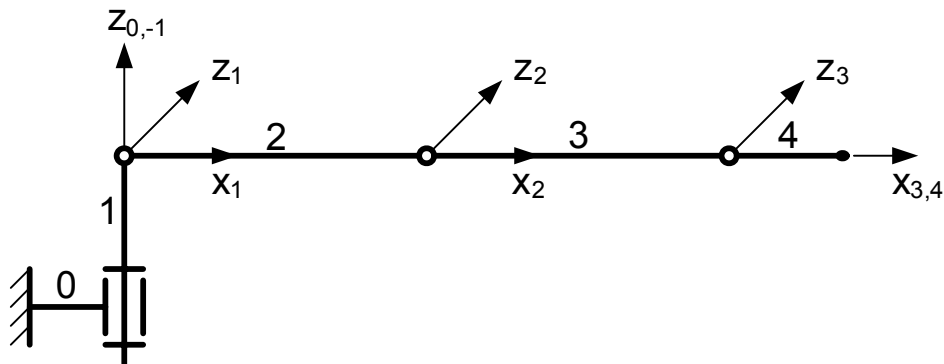


Рис.6. Кинематическая схема четырехзвенного ангулярного робота

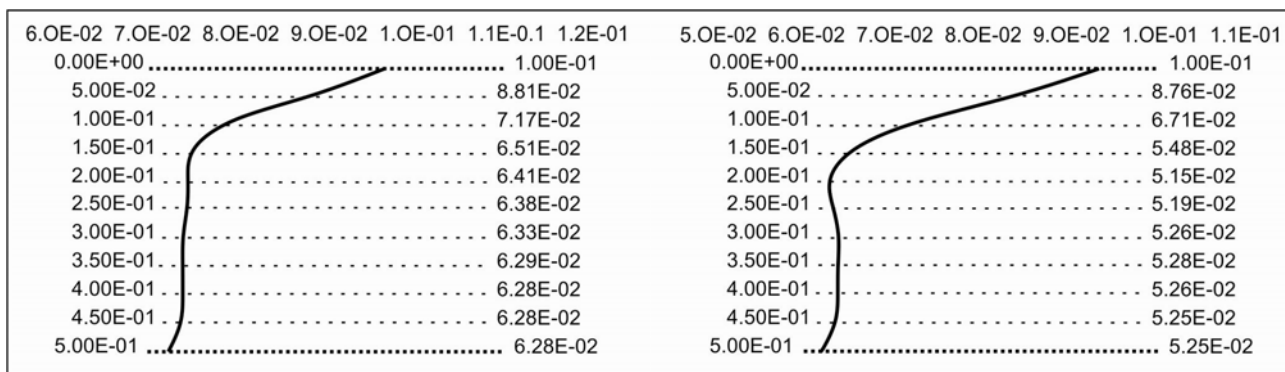


Рис. 7. Аперiodические переходные процессы перемещения по высоте конечного звена манипулятора

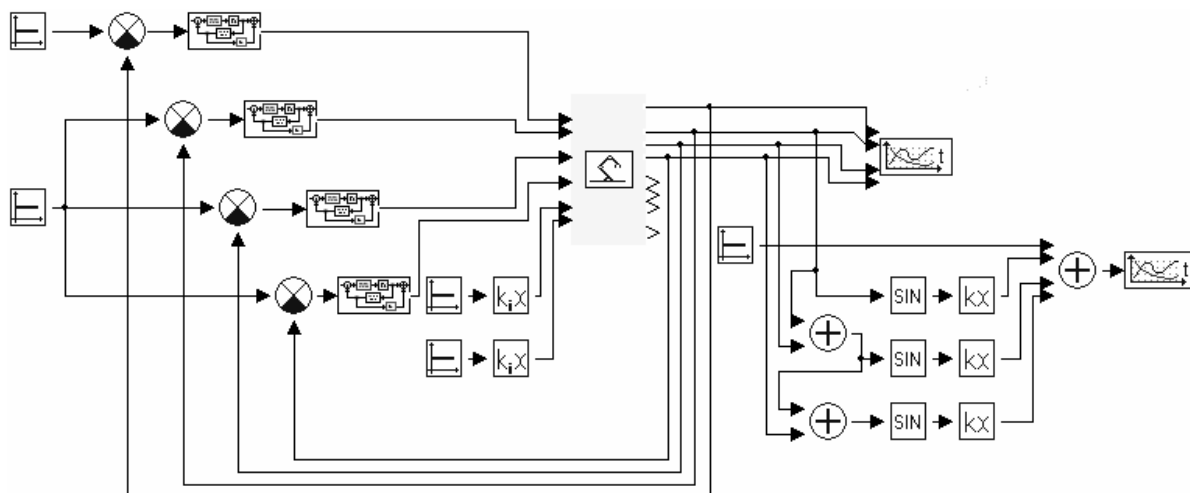


Рис.8. Блок четырехзвенного манипулятора в пакете “МВТУ-3.7”

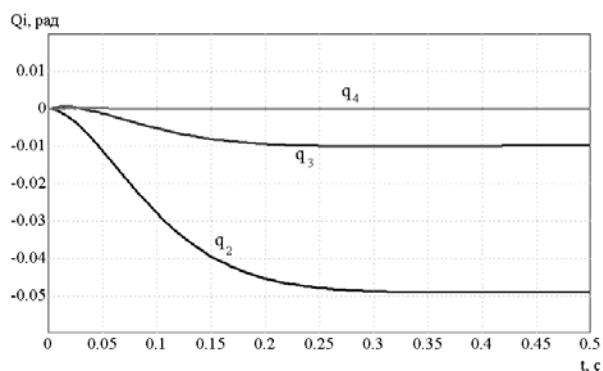


Рис. 9. Переходные процессы суставных координат

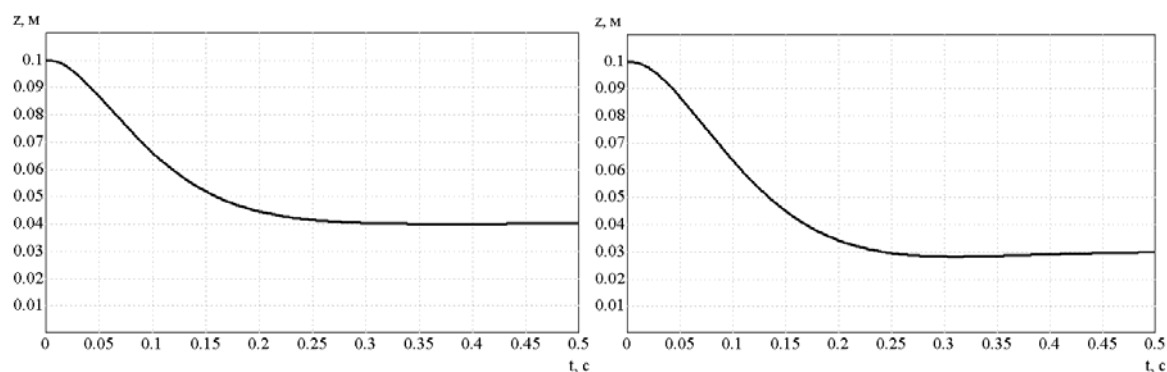


Рис. 10. Переходной процесс перемещения конечного звена

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ МНОГОСУСТАВНЫХ РОБОТОВ В ПРОГРАММНОМ КОМПЛЕКСЕ «УНИВЕРСАЛЬНЫЙ МЕХАНИЗМ» НА ПРИМЕРЕ МАНИПУЛЯТОРА РМ-01.

Для моделирования движения манипулятора далее использовался программный комплекс «Универсальный механизм» (УМ), предназначенный для автоматизации процесса исследования механических объектов, которые могут быть представлены системой абсолютно твердых или упругих тел, связанных посредством кинематических и силовых элементов.

Дерево-структура описания модели и внешний вид РМ-01 в «Универсальном механизме» изображены на рис. 11.

Робот состоит из шести тел (звеньев), представленных шестью графическими образами. Также были смоделированы рабочий инструмент и рабочая поверхность (сфера на «подставке»). Основные геометрические параметры манипулятора РМ-01 известны [1]. Массовые характеристики и диагональные элементы тензоров инерции звеньев приведены ниже в табл. 3 [3]. Масса и элементы тензора инерции первого звена относительно осей Ох, Оу не важны, поэтому они были выбраны произвольно.

Таблица 3

Звено	I_{xx}	I_{yy}	I_{zz}	Масса, кг
1	-	-	0,35	-
2	0,13	0,524	0,539	17,4
3	0,066	0,0125	0,086	4,8
4	$1,8 \times 10^{-3}$	$1,8 \times 10^{-3}$	$1,3 \times 10^{-3}$	0,82
5	$0,3 \times 10^{-3}$	$0,3 \times 10^{-3}$	$0,4 \times 10^{-3}$	0,34
6	$0,15 \times 10^{-3}$	$0,15 \times 10^{-3}$	$0,04 \times 10^{-3}$	0,09

Модель системы управления для данной задачи создается в среде MATLAB-Simulink, затем импортируется в УМ и связывается с моделью механической части, что в итоге позволяет проанализировать динамику управляемого движения.

В настоящей работе реализована классическая модель позиционного кинематического управления в два этапа. На первом этапе планируется траектория, т.е. предварительно определяется программное движение степеней подвижности на некотором временном отрезке, на втором – непосредственно отрабатывается полученная программная траектория приводами робота. При массаже необходимо отслеживать положение и ориентацию рабочего органа манипулятора в пространстве координат схвата. Программная траектория планируется сначала в декартовых координатах, затем решением обратной задачи кинематики (ОЗК) получается программная траектория в пространстве обобщенных координат. В нашем случае это возможно, так как ОЗК для манипулятора РМ-01 решается аналитически.

Математическая модель привода для двигателя постоянного тока с независимым возбуждением и управлением по напряжению на якоре имеет известный вид:

$$\begin{cases} u = iR + C_\omega \dot{\phi}, \\ M = C_m i. \end{cases} \quad (2)$$

Здесь i – ток в цепи якоря двигателя, C_m – коэффициент электромагнитного момента, C_ω – коэффициент противо-э.д.с. двигателя, R – активное сопротивление цепи якоря двигателя, M – момент двигателя.

Из формулы (2) следует выражение для момента двигателя через напряжение на якоре и угловую скорость вращения вала:

$$M = b_1 u - b_2 \dot{\phi},$$

$$\text{где } b_1 = \frac{C_m}{R}, b_2 = \frac{C_m C_\omega}{R}.$$

Для формирования управляющих сигналов, чтобы изучить точность и качество переходных процессов, в нашей задаче используется ПД-регулятор:

$$u_i = -k_{i1} \Delta \varphi_i - k_{i2} \Delta \dot{\varphi}_i,$$

где $\Delta \varphi_i = \varphi_i - \varphi_{inp}$ – рассогласование программной и реальной траекторий, $i = 1, \dots, 6$.

В MATLAB-Simulink вышеуказанный привод с ПД-регулятором задается блок-схемой (см. рис. 12).

Для моделирования контактного взаимодействия в УМ используется силовой элемент типа «контакт сфера-сфера», реализующий модель податливого контакта, при котором допускается внедрение контактирующих элементов одного тела в поверхность, связанную с другим телом. «Точка контакта» определяется путем минимизации расстояния между поверхностями (например, между сферой и плоскостью, сферой и Z-поверхностью и т. п.) [7].

В работе используется следующая модель контактных сил: нормальная сила определяется в соответствии с моделью вязкоупругого взаимодействия, касательная сила вычисляется в соответствии с моделью сухого трения –

$$\mathbf{N} = -(c\Delta z + k\Delta \dot{z})\mathbf{v}, \mathbf{F} = \begin{cases} -\frac{fN\mathbf{v}}{|\mathbf{v}|}, & |\mathbf{v}| > v_f, \\ -\frac{fN\mathbf{v}}{v_f}, & |\mathbf{v}| \leq v_f, \end{cases}$$

где $\mathbf{N}$ – нормальная реакция, $\mathbf{F}$ – сила сухого трения, $\mathbf{v}$ – нормаль к поверхности, Δz – глубина внедрения, c, k – параметры поверхности, $\mathbf{v}$ – скорость проскальзывания, f – коэффициент трения, v_f – достаточно малая величина.

Для задачи моделирования массажа было реализовано два вида взаимодействия рабочего инструмента робота с моделью рабочей поверхности:

- модель взаимодействия, при котором инструмент погружается в мягкую ткань бесконечной толщины;

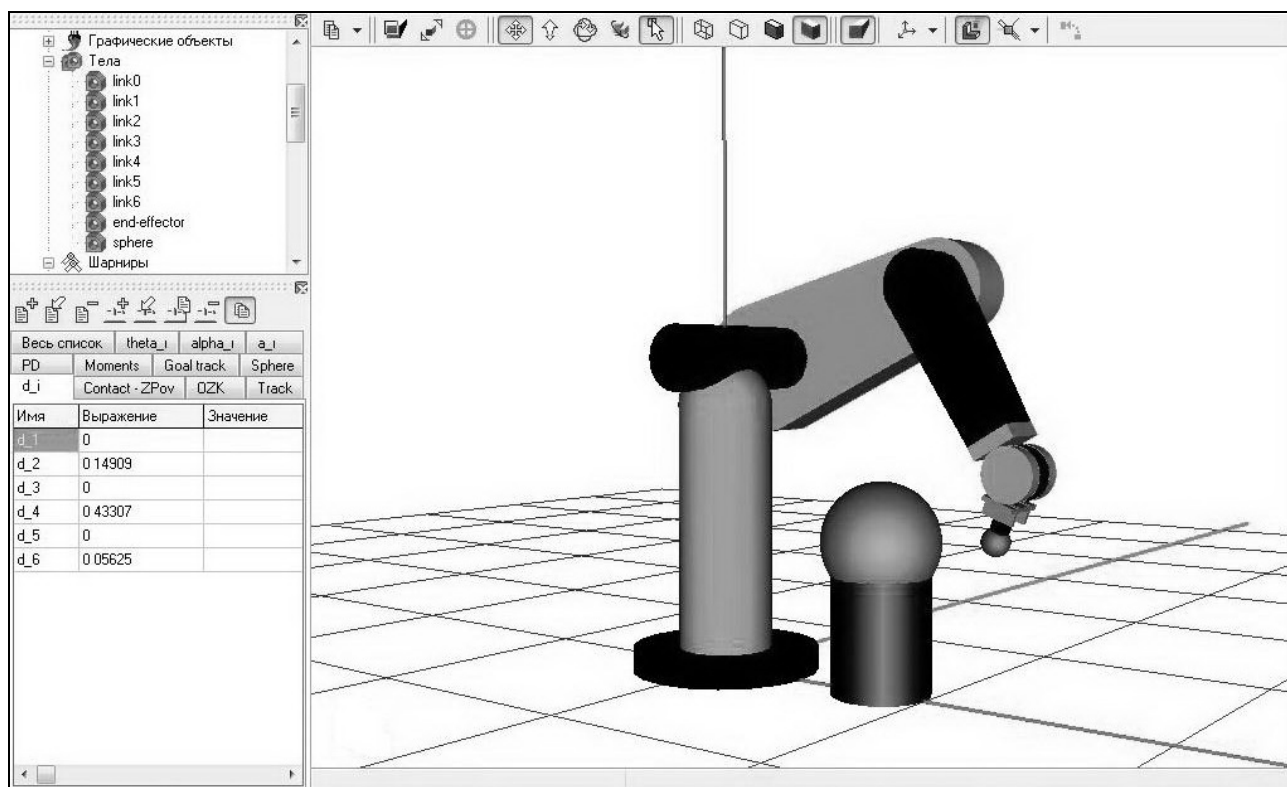


Рис. 11. Дерево-структура и общий вид РМ-01 и рабочей поверхности в UM

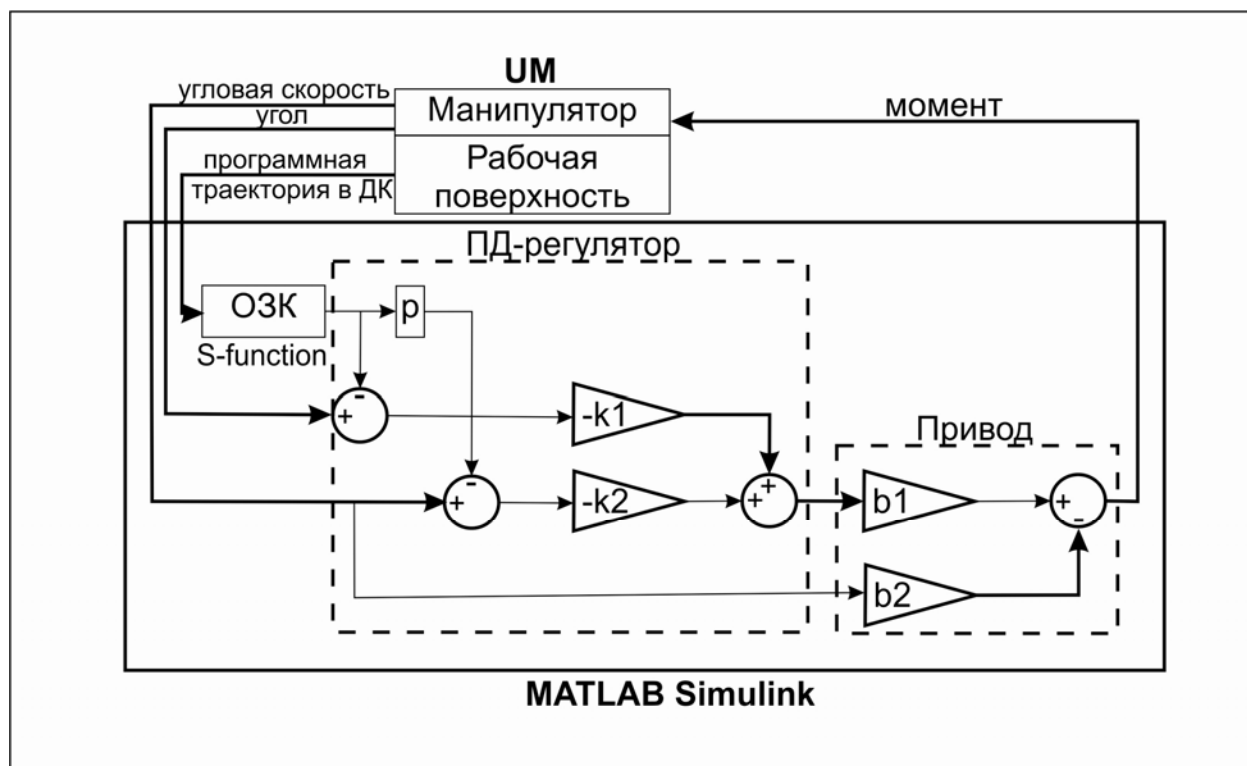


Рис. 12. Блок-схема ПД-регулятора и привода в MATLAB Simulink

- модель взаимодействия (с «подложкой»), при котором мягкая ткань ограничена с одной стороны твердой поверхностью (костной тканью).

Первая модель реализована с помощью стандартного инструмента УМ: силовой элемент типа «контакт сфера – сфера». Вторая – использует тот же инструмент УМ, но дважды. Для этого создаются два тела, вложенных друг в друга. Первое тело, полое, играет роль мягкой ткани (толщина оболочки совпадает с необходимой толщиной мягкой ткани), второе, сплошное, играет роль костной ткани.

Для описанной модели были получены следующие результаты.

В поставленных экспериментах рабочий инструмент движется по траектории, описывающей дугу окружности, при разной глубине погружения h в рабочую поверхность и разных параметрах привода.

Параметры контактного взаимодействия:
 $c = 10000 \text{ Н} / \text{м}, k = 100 \text{ Нс} / \text{м}, f = 0,25 \text{ м}.$

Далее приведены результаты экспериментов при $h = 2 \text{ мм}.$

Параметры приводов:

$$b_{i1} = 4000, b_{i2} = 1000, i = 1, \dots, 6.$$

$$k_{11} = 100, k_{12} = 0.2, k_{21} = 100, k_{22} = 0.17, k_{31} = 100, k_{32} = 0.14,$$

$$k_{41} = 90, k_{42} = 0.11, k_{51} = 90, k_{52} = 0.08, k_{61} = 90, k_{62} = 0.05.$$

Ниже проиллюстрирована раскадровка экспериментов (рис. 13):

На рис. 14 и 15 представлены реальные траектории и скорости в сочленениях манипулятора.

Здесь a,b,c,d,e,f – траектории и скорости в сочленениях соответственно для 1-6 углов и далее аналогично для моментов.

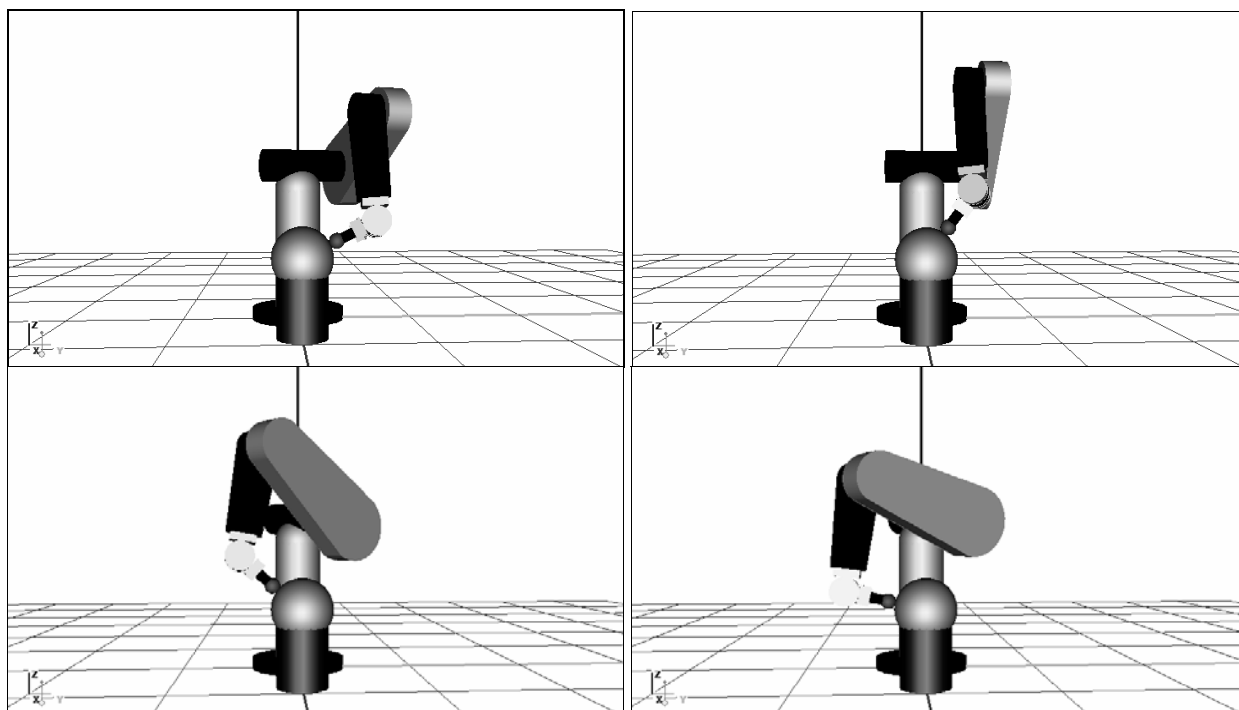


Рис. 13. Раскадровка экспериментов

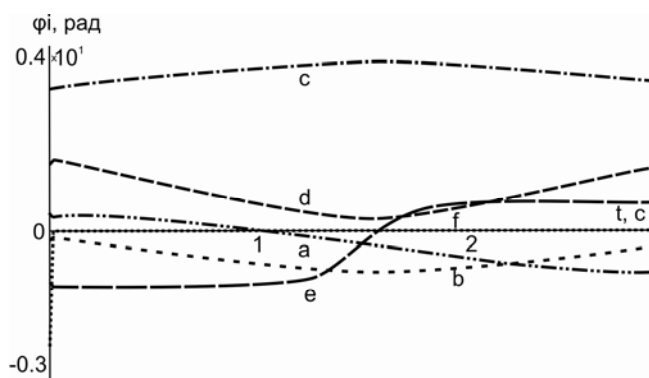


Рис. 14 График угловых траекторий в сочленениях

На последнем рисунке видны биения в начале движения, которые обусловлены качеством переходных процессов при выходе на начало программной траектории. Более наглядно данный эффект заметен на следующем графике моментов, развиваемых приводами для реализации программной траектории (см. рис. 16).

График на рис. 17 показывает величину контактных сил при движении рабочего органа вдоль траектории с погружением в рабочую поверхность на 2 мм. Здесь a , b , c – величины главного вектора кон-

тактных сил, нормальной силы и силы трения. Видно, что при данном управлении и параметрах приводов удается сохранять постоянными величины контактных сил на всей траектории.

Таким образом, моделирование в UM позволяет совмещать удобство реализации управления в виде блок-схем (интеграция с MATLAB Simulink), как это делается в MBТУ 3.7 и «Робот», с графическим пакетом и широким ассортиментом инструментов для анализа динамики и кинематики самого UM.

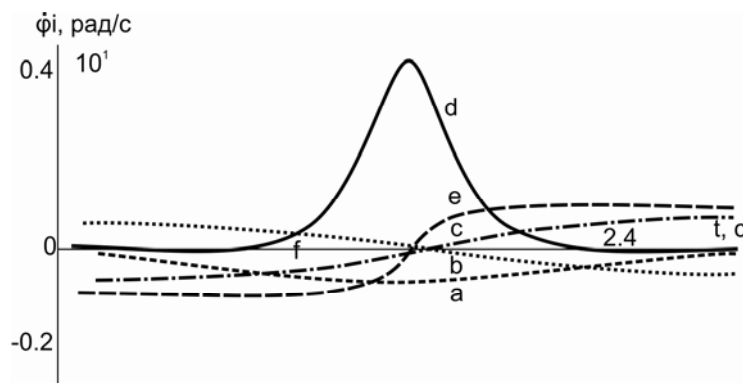


Рис. 15 График угловых скоростей в сочленениях

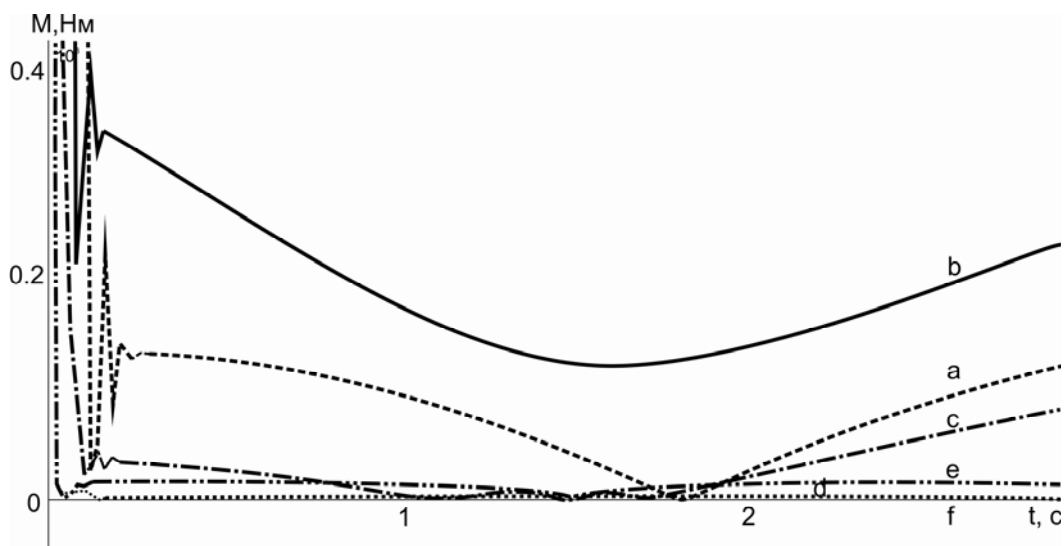


Рис. 16. График моментов в приводах манипулятора

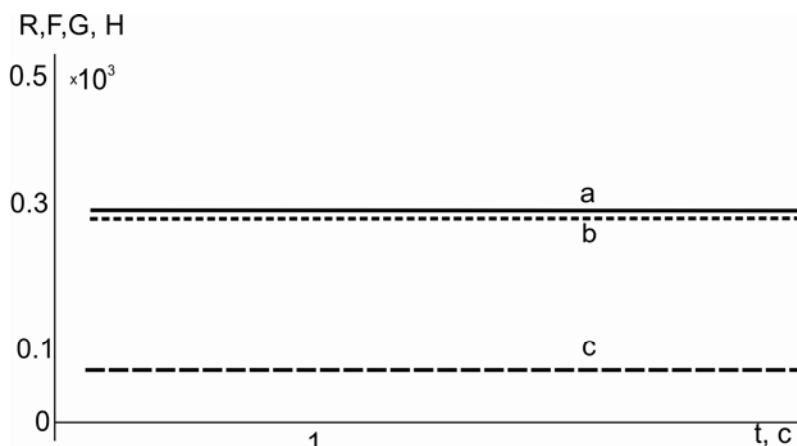


Рис. 17 График контактных сил

ВЫВОДЫ

Проведенные эксперименты позволяют сделать следующий вывод. Несмотря на наличие в механотерапии сложных сред с распределенными параметрами (МТ представляет собой сплошную среду), для эффективного моделирования робота можно применять введенные в работе модели взаимодействия и структурные компоненты роботов и использовать средства твердотельного моделирования. Такие модели оказываются существенно более эффективными в вычислительном смысле, но дают достаточно богатое и адекватное описание рассмотренных задач. Предпочтителен в этом подходе комплекс «Универсальный механизм».

Более подробное моделирование приводит к выводу, что идеальный многозвенный робот для массажа – легкая рука (привод - безредукторный, малоинерционный), она может выполнять «поглаживание» с контактным усилием 1 Н и она же может выполнять «разминание» с усилием 100 Н при резком рельефе рабочей поверхности.

СПИСОК ЛИТЕРАТУРЫ

1. Фу К., Гонсалес Р., Ли К. Робототехника. - М.: Мир, 1989. – 624 с.
2. Зенкевич С.Л., Ющенко А.С. Основы управления манипуляционными роботами: учебник для вузов. 2-е изд., исправ. и доп. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 480 с.
3. Armstrong B., Khatib O., Burdick J. The explicit dynamic model and inertial parameters of the PUMA 560 arm // Proc. IEEE Int. Conf. Robotics and Automation. Proceedings. – 1986. – Vol. 3. – P. 510 – 518.
4. Юревич Е.И. Основы робототехники. 2-изд., перераб. и доп. – СПб.: БХВ-Петербург, 2005. – 416 с.
5. Журавлев В.В. Система позиционно силового управления для механотерапии: дис. ... канд. техн. наук. – М., 2011. – 144 с.
6. Филаретов В.Ф., Юхимец Д.А., Мурсалимов Э.Ш. Особенности использования программной среды MatLab для визуального моделирования движений многозвенных манипуляторов // Мехатроника, автоматизация и управление. – 2010. – № 9. – С. 66 – 71.
7. Universal Mechanism. – URL: <http://umlab.ru/>
8. MatLab. – URL: <http://www.mathworks.com/>

9. Golovin V.F., Grib A.N. Mechatronic system for manual therapy and massage. // Proc. of 8-th Mechatronics Forum International Conference, University of Twente. – Netherlands. – 2002.
10. Golovin V.F., Grib A.N. Computer assisted robot for massage and mobilization. // Proc. “Computer Science and Information Technologies”. – Conference Greece University of Patras. – 2002.
11. Джеймс Ф. Йонг. Основные методы управления - управление роботами. - Университет Райс, Хьюстон, штат Техас, США. – URL: <http://www.owlnet.rice.edu/~elec201/Book/control.html>
12. Голубев Ю.Ф. Основы теоретической механики: учебник. 2-е изд., перераб. и дополн. – М.: Изд-во МГУ, 2000 г., – 720 с.
13. Погорелов Д.Ю. О численных методах моделирования движения систем твердых тел. // Журнал вычислительной математики и математической физики. – 1995 – № 4. – С. 501-506.

Материал поступил в редакцию 09.10.2012.

Сведения об авторах

ГОЛОВИН Вадим Фёдорович – кандидат технических наук, доцент Московского государственного промышленного университета (ФГБОУ ВПО «МГИУ»)

E-mail: medicalrobot@mail.ru

АРХИПОВ Максим Викторович – ассистент лаборатории «Робототехника» кафедры Автоматика, информатика и системы управления ФГБОУ ВПО «МГИУ»

E-mail: medicalrobot@mail.ru

ЖУРАВЛЁВ Виталий Валерьевич – кандидат технических наук, ассистент лаборатории «Робототехника» кафедры Автоматика, информатика и системы управления ФГБОУ ВПО «МГИУ»

E-mail: medicalrobot@mail.ru

ПАВЛОВСКИЙ Владимир Евгеньевич – доктор физико-математических наук, профессор, главный научный сотрудник Института прикладной математики им. М.В.Келдыша РАН, Москва

E-mail: vlpavl@mail.ru

ОРЛОВ Игорь Александрович – аспирант Института прикладной математики им. М.В. Келдыша РАН, Москва

E-mail: orlovbel@gmail.com

Организация сбора данных систем спутникового позиционирования в сети стационарных приемных пунктов*

Описан опыт создания системы, позволяющей осуществлять сбор и обработку данных систем спутникового позиционирования с сети стационарных станций для построения модели ионосферы и дальнейшего анализа. Основной акцент сделан на организации хранения данных в реляционной СУБД (системе управления базы данных) в формате, дающем возможность интеллектуального анализа хранимых данных, а также на создании распределенной сети, объединяющей информацию с отдельных станций, для обеспечения геодезических сервисов и анализа ионосферы. Автоматический анализ документации формата компании JAVAD и построение метайнформации о формате сделал возможным хранение данных в СУБД в значимом виде и создание программного фреймворка для интеллектуального анализа этих данных. Проблема хранения большого объема данных со станций решается распределенной архитектурой, позволяющей легко организовать хранение данных как на серверном кластере или кластерах, так и в облаке.

Ключевые слова: сбор данных, анализ естественных текстов, метапрограммирование, JAVAD, GNSS

ВВЕДЕНИЕ

Применение спутниковых технологий основано на получении данных с систем спутникового позиционирования (далее по тексту все системы спутникового позиционирования будут называться аббревиатурой *GNSS* - *Global Navigation Satellite System*). Кроме очевидного использования таких систем в геодезии, картографии, навигации и мониторинга транспорта, размещение достаточно плотной сети регистрирующих станций позволяет фиксировать тектонические движения (см., например, [1]), и исследовать свойства ионосферы [2, 3].

В настоящее время имеется несколько наземных сетей станций наблюдения, наиболее крупные из которых расположены в Северной Америке (объединенные под эгидой *National Geodetic Survey* [4]) и Западной Европе (*EUREF Permanent Network (EPN)* [5]). Эти сети, наряду с еще более чем 200 агентами, входят в Международную службу систем спутникового позиционирования (*The International GNSS Service* [6]).

В настоящее время в Российской Федерации также проводится ряд мероприятий, направленных на построение такого рода сетей, в том числе и для мониторинга ионосферы [7]. Примерный объем трафика при ежесекундной частоте опроса составит 1 гигабайт в сутки на станцию. При планируемом размещении нескольких сотен пунктов регистрации, организация хранения и обработки данных потребует серьезных ресурсов. Кроме того, возникает комплекс задач по организации безопасного и надежного кана-

ла передачи данных в режиме реального времени, хранению данных в центре обработки с последующей организацией доступа к собранным данным для их анализа и обработки. Очевидно, что тривиальные решения, предполагающие хранение данных в виде файлов и использование высокоуровневых протоколов, в данном случае неприемлемы.

В настоящей работе предложен вариант организации хранения данных, согласованный с процедурой ввода данных с пункта регистрации. Последний представляет собой *GNSS*-приемник компании *JAVAD*, дополненный персональным компьютером под управлением операционной системы *Windows 7*. Приемник обладает набором стандартных интерфейсов для связи с компьютером и предоставляемым производителем программным обеспечением, позволяющим получать данные в виде файлов формата *JPS (JAVADGREIS)* [8] и *BINEX (Binary Exchange Format)* [9].

Нами представлено решение, основанное на использовании средств реляционной СУБД, дополненное набором разработанных программных средств, служащих для ввода-вывода данных и их анализа. Безопасный и надежный канал передачи данных организуется средствами самой СУБД. Ввод данных в базу и их предварительная обработка осуществляется на пункте регистрации, что уменьшает нагрузку на центр обработки. Данные хранятся в структурированном и готовом к анализу виде. Последующий доступ к собранным данным осуществляется обращением к базе с помощью утилит, разработанных на основе специально созданной библиотеки, инкапсулирующей всю сложную работу в простые манипуляции объектной моделью.

* Работа выполнена при частичной поддержке гранта РФФИ 10-07-00491-а

ФОРМАТ

При получении данных мы имеем дело с форматами производителей приемного оборудования. В нашем случае речь идет о формате *JPS*, являющемся основным для используемых нами устройств-приемников фирмы *JAVAD*. При обмене *GNSS*-данными используются стандартные форматы *RINEX* (*Receiver Independent Exchange Format*) [10] и *NTRIP* (*Network Transport of RTCM via Internet Protocol*) [11]. Стандартные форматы являются более узкими и не могут содержать все данные, получаемые с конкретного регистратора. Таким образом, возникает задача организации получаемых данных в форме, допускающей как достаточно простую их первичную обработку, так и обмен этими данными. При выборе формата мы исходили из его полноты описания получаемых данных. Поэтому, в качестве основы структуры хранилища данных был принят формат производителя, в нашем случае это *JPS*, позволяющий, при необходимости, осуществлять вывод данных из системы как в максимально полном исходном формате *JPS*, так и в более общих форматах *RINEX* или *NTRIP*.

Еще одним существенным вопросом является степень глубины анализа данных. Следует ли результаты измерений хранить в виде семантически значимых значений, или достаточно работать с данными как с массивом байт? Первое позволяет нам исследовать полученные данные, осуществлять их поиск в системе по важным для анализа критериям, что превращает систему в нечто большее, чем просто средство сбора и хранения большого объема абстрактных данных из множества источников – в серьезный инструмент постобработки. Однако это требует серьезной работы с точки зрения анализа формата и преобразования входных данных в массив имеющих физический смысл значений. В случае, когда формат данных является тривиальным, такая работа сводится к созданию вручную некоторой структуры базы данных, отражающей их физический смысл, и написанию конвертера из исходного формата в эту структуру. Формат *JPS* представляет собой поток данных, состоящий из более сотни различных типов сообщений, кодирующих в бинарном представлении определенный массив данных. Поэтому задача создания конвертера данных из этого формата не является тривиальной и исключает «ручную» обработку.

Описание формата *JPS* содержится в документе *JAVAD GNSS Receiver External Interface Specification (JAVAD GREIS)*. Ключевой особенностью этого документа, повлиявшей на реализацию системы, является его строгость и достаточно высокое качество. Для описания сообщений в нем принят Си-подобный синтаксис, описывающий структуру сообщения, который дополняется словесным описанием особенностей, которые сложно передать декларативно, например, фактический размер динамических полей. Высокое качество в данном случае заключается в единообразии, точности и полноте описания всех типов сообщений. Стоит заметить, что каждое сообщение имеет фиксированные поля с идентификатором и размером, что позволяет организовать обработку лишь части сообщений в случае необходимости.

Описанные выше особенности привели нас к решению организовать автоматическую обработку этого формата данных. Результатом такой обработки является метаданная, достаточная для полного анализа всех сообщений формата, что подразумевает наличие информации о структуре каждого сообщения до уровня отдельных полей данных.

[SI] Satellite Indices

```
struct SatIndex {nSats+1} {  
    ul usi[nSats]; // USI array []  
    ul cs;         // Checksum  
};
```

Рис. 1. Фрагмент описания сообщения в документации *GREIS*. Из фрагмента, помимо описания структуры, можно выделить комментарии к каждому полю («*USIarray []*» ит.д.), название (*Satellite Indices*), *ASCII* код сообщения (*SI*). Текст «*{nSats+1}*» указывает на динамический (но вычисляемый простым алгоритмом) размер структуры.

Для выделения метаданной был применен автоматический анализ текста документации. Приведем краткое описание этого анализа. Его можно разделить на достаточно традиционные для данного типа задач этапы:

1. Поиск и выделение в документации интересующих фрагментов текста (подобно фрагменту, приведенному на рис. 1).
2. Лексический/синтаксический анализ выделенных фрагментов текста.
3. Семантический анализ полученного синтаксического дерева.

Исходная документация преобразуется в простой текстовый формат и анализируется с целью выявления фрагментов, схожих с приведенным выше. Для их выявления используется также общая информация о документе, такая как оглавление, позволяющая получить список из всех имеющихся в документации сообщений и их примерное месторасположение в тексте.

Полученные интервалы преобразуются в синтаксическую структуру, которая обладает всей необходимой информацией, кроме размера динамических полей.

Размер полей (и структур сообщений в целом) можно условно разделить на фиксированный и динамический равномерно распределенный, определяющийся сложным алгоритмом.

Динамический равномерно распределенный размер полей подразумевает, что структура сообщения имеет фиксированный размер типов данных у полей, однако размер массива элементов ряда полей неизвестен и определяется по формуле:

$$AS = \frac{MS - FFS}{DFTS},$$

где: *MS* – фактический размер поступающей на обработку структуры, *FFS* – суммарный размер полей с фиксированной длиной, *DFTS* – сумма размеров типов данных у полей с динамической длиной.

Для сообщений, размер которых невозможно определить автоматически, требуется явная реализация алгоритма определения размера.

Метаинформация, получаемая в результате обработки, содержит не только информацию, необходимую для описания структуры сообщений, но также извлеченные комментарии и краткое текстовое описание этих сообщений, что дает возможность документирования любого программного кода, сформированного на основе данной метаинформации, а также позволяет создавать внешние справочники с более структурированным описанием используемого формата. Полученная метаинформация хранится в формате *XML (Extensible Markup Language)*.

Поскольку используемый компанией *JAVAD* стиль документирования является широко распространенным и общепризнанным, примененный подход позволяет охватить не только формат *JAVAD*, но и другие промышленные стандарты.

Когда получена метаинформация о формате *JPS*, становится возможным создание программного обеспечения для работы с ним, включая средства генерации структуры базы данных и утилит занесения данных в систему. Более подробно эти вопросы раскрыты далее.

РАСПРЕДЕЛЕННАЯ БАЗА ДАННЫХ

При разработке общей архитектуры созданной системы требовалось учесть сразу несколько факторов. Ряд проблем, таких как организация канала передачи данных и прозрачный доступ к данным со стороны пользователя, был решен использованием в качестве основы нашей инфраструктуры реляционной СУБД *MySQL*. Перед нами, однако, стоял вопрос масштабирования базы данных (БД) с целью обработки большого объема данных. Масштабирование базы данных возможно несколькими путями. С одной стороны, можно использовать стандартные средства *MySQL*, такие как разбиение на разделы и репликация [12], которые позволяют распределить нагрузку на базу данных по нескольким серверам. С другой стороны, возможно решить этот вопрос на уровне структуры базы данных и хранить данные с разных станций регистрации в отдельных базах.

При использовании единой базы данных для всех пунктов наблюдения проблемы масштабирования системы решаются при помощи стандартных средств *MySQL*, таких как разбиение на разделы и репликация. Хранение всех данных в одной базе позволяет при необходимости осуществлять кросс-выборку данных с разных станций. При этом управление правами доступа пользователей реализуется элементарно. Однако этот подход требует построения более сложной структуры базы данных, которая позволит вместить данные со всех регистраторов, а также более сложной организации информационно-технологической (ИТ) инфраструктуры в дата-центре. Усложнение структуры базы данных отрицательно сказывается на ее читаемости и, что важнее, на производительности системы. При этом значительно усложняется модификация ИТ-инфраструктуры, которая потребует, например, при изменении конфигурации сети станций наблюдения. Наконец, использование единой базы данных делает невозможным разграничение прав доступа к данным с отдельных станций средствами СУБД.

Вариант с хранением данных с разных станций регистрации в отдельных базах нам представляется предпочтительным, поскольку позволит существенно упростить масштабирование, а при необходимости сделать центр обработки данных территориально распределенным. Появляется также возможность разграничения доступа к отдельным станциям на уровне пользователей базы данных. Недостатки, такие как более сложное управление правами пользователей, невозможность осуществления кросс-выборки данных с разных станций в одном *SQL*-запросе, усложнение структуры из-за необходимости доступа к информации о существующих станциях-базах, легко снимаются написанием простых скриптов, либо (говоря про кросс-выборку) являются незначительными. Простота масштабирования и распределения ресурсов с лихвой окупает указанные выше затраты.

Основная работа при использовании этого подхода ложится на программную составляющую, объединяющую все источники данных (базы данных) (рис. 2).

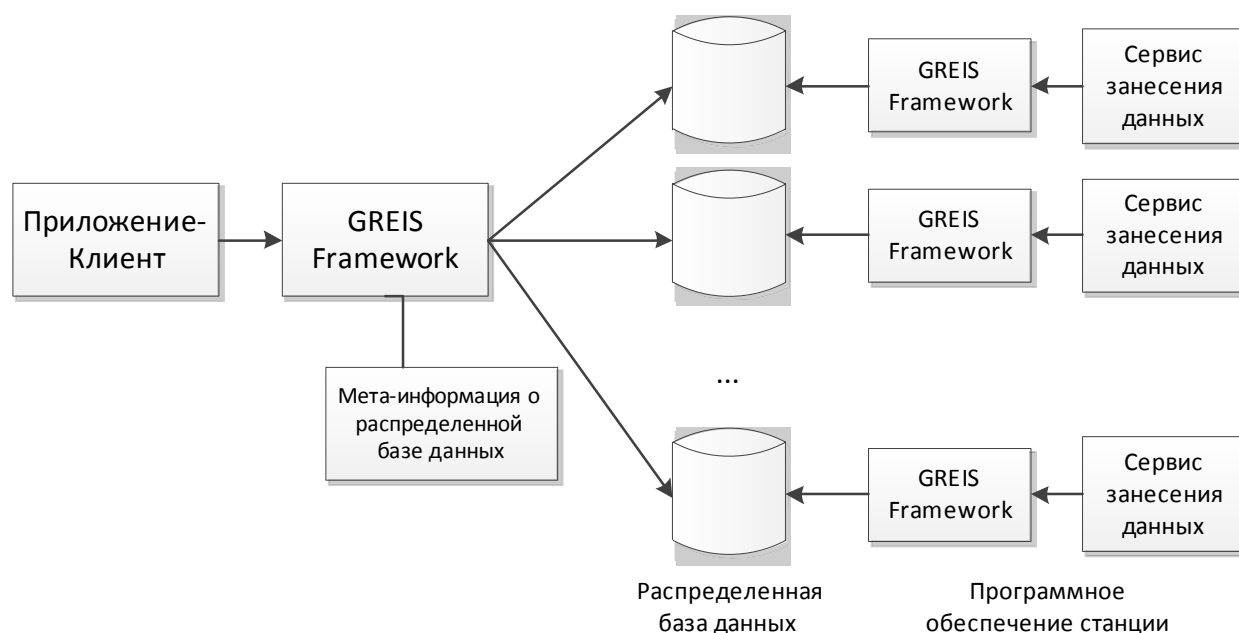


Рис. 2. Общая архитектура созданной системы. Ключевые узлы системы – распределенная база данных и библиотека *GREIS*, предоставляющая *API (Application Programming Interface)* для ввода, вывода и анализа данных. Под каждую станцию выделяется отдельная база данных. Программное обеспечение на станции осуществляет ввод данных в БД. Приложение-клиент производит анализ данных или их вывод из системы в одном из поддерживаемых форматов (*JPS, RINEX, NTRIP*).

Создаваемая на основе метainформации структура базы данных позволяет организовать хранение сообщений *JPS* как максимально развернуто, на уровне явного представления примитивных типов, так и в бинарном виде, как максимум, сортируя сообщения по их типу. Требуется найти определенный баланс, при котором структура, с одной стороны, позволяет проводить анализ данных по важным критериям, а, с другой стороны, хранит незначимые данные в бинарном виде, ускоряя, таким образом, время обращения к ней.

Основной поток сообщений несет значимую информацию, и потому в нашей реализации он хранится в максимально открытом виде. В приведенном на рис. 3 фрагменте физической модели в качестве примера представлены сгенерированная сущность *GPS*-эфемерид и являющаяся ее частью сущность сложного типа данных. Каждое сообщение и сложный тип данных представлены в виде отдельных таблиц, содержащих в явном виде физически значимые значения. Сложные типы данных хранятся в отдельных сущностях и связываются по внешнему ключу.

К сообщениям, хранящимся в бинарном виде, в основном можно отнести не несущие важной для анализа информации сообщения, например, журна-

лирование событий приемного устройства. Кроме того, в бинарном формате хранятся сообщения, слишком сложные для полного разбора по причине наличия полей с динамическим размером, определяемым сложным алгоритмом. Несмотря на то, что некоторые из таких сообщений содержат значимую информацию, из-за сложностей в реализации алгоритма их разбора предпочтительнее хранение таких сообщений в бинарном виде. Кроме того, для увеличения быстродействия в бинарных полях хранятся также массивы примитивных типов.

Генерация структуры базы данных осуществляется с помощью автоматически созданных *SQL*-сценариев генерации структуры. Однако такой уровень документирования не позволяет организовать доступ к данным, ввиду отсутствия соответствующей метainформации. Поэтому была создана структура таблиц, содержащая метainформацию обо всех сообщениях системы (рис. 4). Таким образом, любое стороннее программное обеспечение потенциально может работать с базой данных без какой-либо посторонней информации и, в том числе, возможно написание своего собственного решения для ввода информации в систему, ее вывода и анализа.

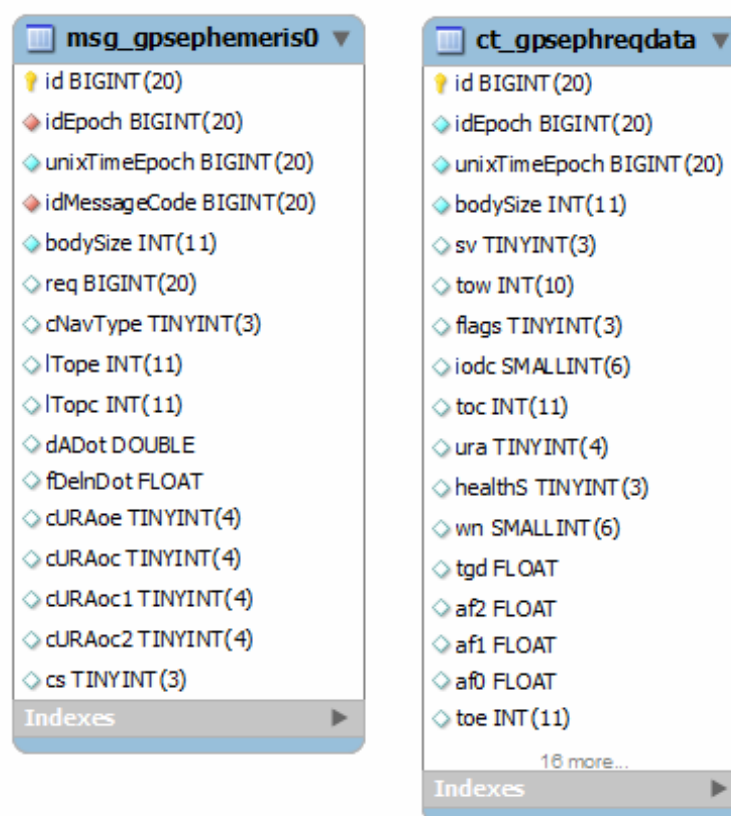


Рис. 3. Сообщения *GPS*-эфемериды как сущность в базе данных. Поле «req» сообщения *msg_gpsephemeris0* представляет сложный тип данных и ссылается на сущность *ct_gpsephreqdata*. Остальные поля представляют простые типы данных и отражают физически значимые значения.

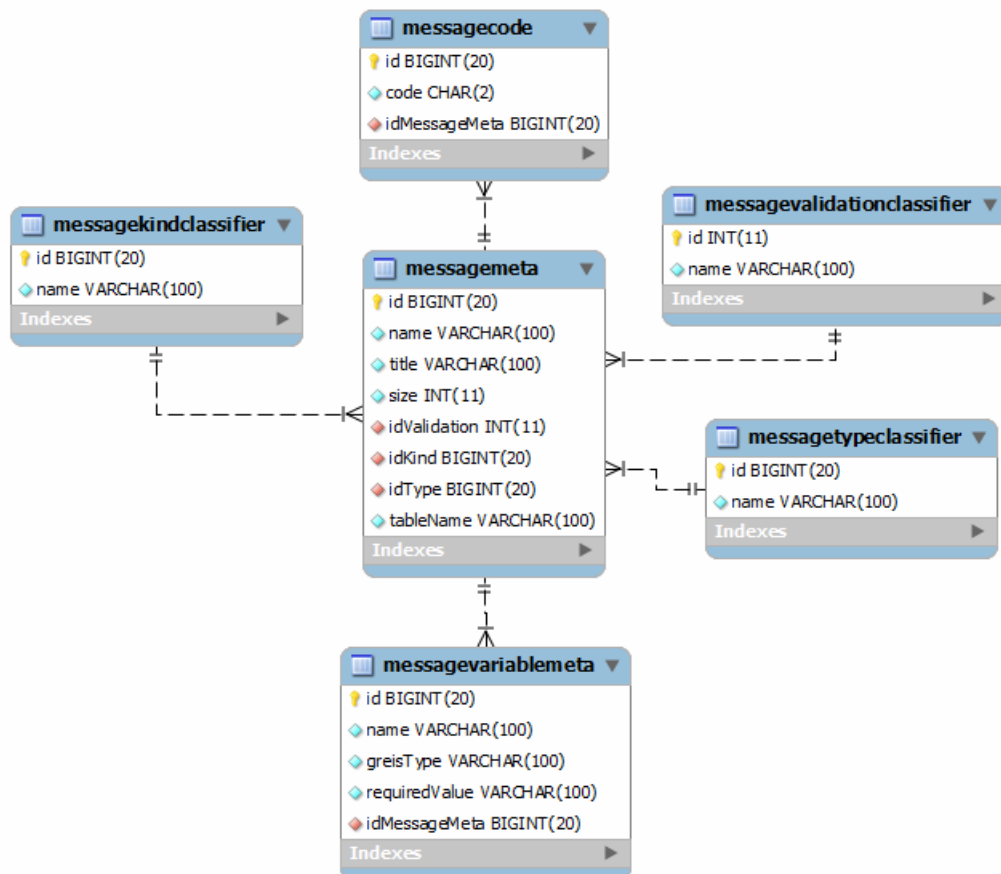


Рис. 4. Часть физической модели, представляющая метainформацию о GREIS-сообщениях. Основные таблицы – *messagemeta* (информация о структурах-сообщениях), *messagevariablemeta* (информация о полях структуры-сообщения), *messagecode* (ASCII-идентификаторы сообщений). Таблицы классификаторы: *messagetype* – тип сообщений (эффемериды, альманахи, простые данные), *messagevalidation* – способ валидации (*checksum*, *crc16*), *messagekind* – способ сериализации сообщений (бинарный, текстовый).

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Сложность внутренней структуры хранения данных подразумевает наличие ряда утилит для ввода данных в систему и их вывода в файлы определенного формата. Очевидно, что в основе таких утилит должна лежать общая библиотека по работе с системой. Эта же библиотека должна предоставлять программный интерфейс (*API*) для более глубокого анализа данных.

Как и в случае с разработкой структуры базы данных, в основе созданной программной библиотеки лежит метainформация. Загрузка метainформации из файла на этапе выполнения позволит изменять логику обработки данных без перекомпиляции всего приложения. Использование же метainформации в качестве основы для генерации кода увеличит быстродействие приложения, однако потребует его перекомпиляции при изменении метainформации. Мы опробовали оба варианта реализации.

Анализ метainформации на этапе выполнения имеет очевидный положительный момент, такой как отсутствие необходимости пересборки системы при изменении спецификации, однако реализация этого подхода выявила целый ряд проблем. Пробная реализация показала такие проблемы, как сложность в разработке, трудности в тестировании системы, а также низкая производительность. Это связано с необходимостью написания большого количества алгоритмически сложного кода, работающего с метainформа-

цией. Помимо сложности, этот код затруднителен в отладке по причине сильной связи с информацией, получаемой на этапе выполнения и, как следствие, уменьшения возможностей компилятора при отслеживании ошибок. Выполнение же подобной логики занимает неоправданно много времени.

Разработка генератора исходного кода, выделяющего всю сложную логику в сам генератор и позволяющего максимально упростить библиотеку для работы с системой, оказалась более удачным решением. Учитывая, что получаемая в результате библиотека будет включать метainформацию на этапе компиляции, большая часть логики, такая как описание типов и сериализация и десериализация в бинарный формат, была вынесена на этап генерации кода. Последнее, в свою очередь, является более простой процедурой. Причиной этому, прежде всего, является возможность проверки сгенерированного кода компилятором. Помимо этого стало возможным использовать более высокоуровневый язык программирования, например, *C#*, избавляющий от таких проблем, как ручное управление памятью. Быстродействие алгоритмов генератора также не является критическим фактором, поскольку этап генерации выполняется лишь единожды при сборке библиотеки.

Получаемый в результате работы генератора программный код (рис. 5) является полностью объектно-ориентированным, обладает избыточной документацией в формате *Doxygen* [13] и позволяет проводить сложный анализ данных.


```

namespace Greis
{
    class GPSEphemeris0StdMessage : public StdMessage
    {
    public:
        SMART_PTR_T(GPSEphemeris0StdMessage);

        GPSEphemeris0StdMessage(const char* p_message, int p_length);
        GPSEphemeris0StdMessage(const std::string& p_id, int p_size);

        Virtual std::string ToString() const;
        virtual std::string Id() const { return _id; }
        virtual EMessageId::Type IdNumber() const { return EMessageId::GPSEphemeris0; }
        virtual bool Validate() const;
        virtual void RecalculateChecksum();
        virtual int BodySize() const { return _bodySize; }
        virtual QByteArray ToByteArray() const;

        ///Required data block
        const GpsEphReqDataCustomType::UniquePtr_t& Req() const { return _req; }
        GpsEphReqDataCustomType::UniquePtr_t& Req() { return _req; }

        /// Signal type nav. data was decoded from
        const Types::u1& CNavType() const { return _cNavType; }
        Types::u1& CNavType() { return _cNavType; }

        /// Time of prediction for ephemeris data
        const Types::i4& LTope() const { return _lTope; }
        Types::i4& LTope() { return _lTope; }

        ...
    }
}

```

Рис. 5. Интерфейс класса GPS-эфемерид. Класс предоставляет доступ к каждому из полей данных, реализует полиморфный интерфейс *StdMessage*, предоставляющий методы подсчета и пересчета контрольных сумм, сериализации и десериализации, определяет типы для упрощения работы с умными указателями C++11 (макрос *SMART_PTR_T*).

Последняя версия библиотеки была реализована в рамках стандарта C++11, что позволило значительно упростить ее реализацию и увеличить быстродействие. Применение *rvalue references* показало значительный прирост производительности при работе с большим количеством небольших объектов, таких как сообщения *JPS*, и упростило использование «умных указателей» (*smart pointers*) и управление памятью. Использование же синтаксического сахара (*syntax sugar*) нового стандарта,

такого как *lambda*-функции, и ключевого слова *auto*, сильно упростило реализацию [14].

Что касается предварительной обработки данных, то следует отдельно отметить вопрос о расположении средств первичного разбора и занесении данных в СУБД. Теоретически такой разбор можно производить как на сервере (в центре обработки), так и на клиенте (регистраторе).

Положительный момент обработки на сервере – наличие значительных вычислительных ресурсов и

отсутствие серьезных ограничений на производительность системы, что позволяет осуществлять разбор крайне высокого уровня сложности. Более того, такой подход является единственно возможным, если станция-регистратор представляет собой устройство слабой производительности, которое неспособно выполнять необходимую обработку данных. Однако минус такого подхода – серьезная нагрузка на центр обработки, что требует выделения в ЦОД (центре обработки данных) дополнительных мощностей и влечет дополнительные расходы.

Разбор на клиенте, в свою очередь, может потребовать от средств обработки повышенного быстродействия и кроссплатформенности, однако дает очевидное преимущество в виде разгрузки центра обработки. Учитывая, что количество станций-регистраторов может превышать сотню, такого рода распределение ресурсов может оказаться весьма существенным.

В разрабатываемой нами системе удалось реализовать разбор данных на клиенте, что позволило воспользоваться вышеописанными преимуществами такого подхода. Ключевым при этом стало наличие на клиенте достаточных вычислительных ресурсов, а также кроссплатформенность разработанной библиотеки, поскольку клиентские машины работают по руководством операционной системы (ОС) *Windows 7*, в то время как на серверах установлена ОС *Linux*.

ЗАКЛЮЧЕНИЕ

Ключевым элементом разработанной системы является сервер с установленной СУБД *MySQL*. Каждой станции соответствует отдельная база данных. Выбранная нами реализация, в зависимости от объема данных, дает возможность хранить данные со всех станций как на одном сервере и обслуживаться одной СУБД, так и на нескольких серверах, в том числе расположенных удаленно друг от друга, в облаке виртуальных машин, предоставляющих хостинг для хранения данных в СУБД *MySQL*. Подобная архитектура позволяет решать проблему хранения большого объема данных и доступа к ним, сводя к минимуму при этом расходы на оборудование в случае небольшого объема данных – большой объем данных может быть распределен на массиве серверов, однако данные могут храниться на одном сервере и обслуживаться одной СУБД для экономии оборудования.

Предварительная обработка данных осуществляется на станции-регистраторе, что позволяет сильно разгрузить вычислительные мощности центра обработки по сравнению с вариантом, когда данные поступают на сервер в необработанном виде.

При регистрации новой станции на сервере создается новая база данных, структура которой формируется посредством автоматически созданного скрипта. На компьютере станции-регистратора устанавливается сервис предварительной обработки данных, подготавливающий и отправляющий данные в базу на удаленном сервере. Таким образом, проблема передачи данных в центр обработки решается на уровне клиент-серверного общения СУБД. Сервис осуществляет мониторинг за поступающими данными в ре-

жиме, близком к реальному времени. Использование сервисом пакетных *SQL*-запросов и возможность использования сжатия данных, а при необходимости и их шифрования, позволяет решать проблемы накладных расходов и безопасности. Проблема накладных расходов является крайне актуальной при использовании традиционных протоколов, таких как *FTP*.

Извлечение данных из системы возможно с использованием специализированной утилиты, сериализующей данные в бинарный формат *JPS (JAVAD GREIS)*. Предоставляемая системой программная библиотека позволяет, в свою очередь, проводить более сложный интеллектуальный анализ данных. Программная библиотека написана на *C++* и является кроссплатформенной. Для подключения новой базы к системе необходимо указать информацию о новой станции в настройках приложения.

Таким образом, используемая система обеспечивает получение и обработку данных с сети распределенных станций и предоставляет конечному пользователю простой и удобный интерфейс для работы с данными, получаемыми в близком к реальному времени режиме.

Опыт применения системы показал ряд узких моментов, требующих дальнейшей проработки. Так, спецификация *JAVAD GREIS* изменяется значительно чаще, чем такие форматы, как *RINEX*. С одной стороны, спецификация, таким образом, уже сейчас включает описания навигационных систем, находящихся еще на этапе развертывания, однако, изменения в спецификации оказались не всегда обратнoсовместимы, что требует организации более сложной структуры базы данных и программного обеспечения для организации прозрачной к этим изменениям работы. Возможность перехода на использование устройств-приемников других производителей также потребует дальнейшего анализа.

* * *

Автор благодарит И.М. Алешина (Институт физики Земли РАН) за постановку задачи и внимание к работе и К.И. Холодкова (Институт физики Земли РАН) за полезные дискуссии.

СПИСОК ЛИТЕРАТУРЫ

1. Motion of the Earth's tectonic plates, the Jet Propulsion Laboratory. – California: Institute of Technology. – URL: <http://sideshow.jpl.nasa.gov/mbh/series.html>
2. Куницын В.Е., Терещенко Е.Д. Томография ионосферы – М.: Наука, 1991. – 176 с.
3. Gary S. Bust and Cathryn N. Mitchell. History, current state, and future directions of ionospheric imaging // Reviews of geophysics. - 2008. - Vol. 46, rg1003. - 23 p.
4. National Geodetic Survey, official website. – URL: <http://www.ngs.noaa.gov>
5. EUREF Permanent Network, official website. – URL: <http://www.epncb.oma.be>
6. The International GNSS Service (IGS), official website. – URL: <http://igsceb.jpl.nasa.gov>

7. Федеральная целевая программа «Создание и развитие системы мониторинга геофизической обстановки над территорией РФ на 2008-2015 гг.»
8. JAVAD GNSS Receiver External Interface Specification (GREIS), March 13, 2012. – URL: http://javad.com/downloads/javadgnss/manuals/GREIS/GREIS_Reference_Guide.pdf
9. BINEX. UNAVCO Geoscience and Geodesy consortium. Format homepage. –URL: <http://binex.unavco.org/binex.html>
10. RINEX. The Receiver Independent Exchange Format Specification. Version 3.01., 2009. – URL: <http://igsb.jpl.nasa.gov/igsb/data/format/rinex301.pdf>
11. Official NTRIP homepage. Bundesamt für Kartographie und Geodäsie (BKG) (Federal Agency for Cartography and Geodesy), Germany. – URL: <http://igs.bkg.bund.de/ntrip/docu>
12. MySQL 5.6 Reference Manual, Chapter 17: Partitioning. – URL: <http://dev.mysql.com/doc/refman/5.6/en/partitioning.html>
13. Doxygen, generate documentation from source. – URL: <http://www.doxygen.org>
14. ISO/IEC 14882:2011 Programming Language C++ (draft). – URL: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf>

Материал поступил в редакцию 10.09.12.

Сведения об авторе

БУРГУЧЕВ Степан Сергеевич - аспирант Института физики Земли им. О.Ю.Шмидта РАН, Москва.
E-mail: sburg@ifz.ru

Автоматическое выделение глагольных актантов: теоретическая основа и актуальные подходы*

Представлен краткий обзор теоретических и практических моделей, используемых при решении задачи автоматической разметки семантических актантов. Рассматривается понятие семантической роли и место этого конструкта в общей языковой системе, проблема составления ролевых инвентарей, а также различные аспекты реализации автоматической разметки актантов с помощью алгоритмов машинного обучения. Также рассматривается ряд проблем, связанных с этой областью, в частности, проблема «языкозависимости» современных методов выделения глагольных актантов.

Ключевые слова: семантические роли, инвентарь семантических ролей, автоматическая разметка актантов, машинное обучение, *semantic role labeling*

1. ВВЕДЕНИЕ

Теория семантических ролей описывает аргументную семантическую структуру предикатов, а также устанавливает соответствия между семантическими аргументами и их конкретными лексико-синтаксическими реализациями. В основе этой теории лежит несколько важных лингвистических наблюдений: наличие корреляции между морфологическими падежами и синтаксическими структурами с одной стороны и семантическими ролями – с другой, а также высокая степень универсальности инвентаря семантических ролей среди языков. Модель, в которой каждый предикат или группа предикатов обладает определенным набором семантических ролей из универсального инвентаря, и при этом каждая роль коррелирует с элементами поверхностного уровня, оказалась привлекательной как в теоретическом, так и в практическом смысле и получила значительное развитие. С тех пор, как теория семантических ролей оформилась в отдельное направление исследований в 70-х гг. XX в., она активно развивалась силами лексикографов и синтаксистов. Будучи проверенной в ходе многочисленных лексикографических и корпусных исследований, теория претерпела изменения, благодаря чему в распоряжении современного исследователя оказывается значительное количество данных о практических аспектах применения теории, а также набор достаточно стабильных моделей для описания предикатной семантики. В рамках этих работ разработаны компактные способы представления лексикографической информации об актантах (например, модели управления), детально исследованы закономерности в кодировании актантов на поверхностном уровне, предложено несколько инвентарей семантических ролей и т.д.

В конце 1990-х – начале 2000-х гг., в связи с увеличением объема текстовых данных, представленных в электронном виде, и ростом доступности компьютеров, теория стала крайне востребована в контексте извлечения информации и автоматической обработки естественного языка, породив отдельное направление – **автоматическую разметку актантов** (Semantic Role Labeling, SRL). Задачу, которая ставится в рамках этого направления, можно кратко охарактеризовать следующим образом: для предиката в тексте на естественном языке требуется провести соответствие между семантическими актантами этого предиката и их реализациями в ситуации, описываемой текстом. Привлекательность такого подхода с инженерной точки зрения состоит в том, что он позволяет при сравнительно небольших затратах создать промежуточный семантический уровень, которого зачастую достаточно для решения конкретных прикладных задач. Например, в задаче извлечения данных (data mining) очень редко требуется глубинный анализ семантики предложенного текста, и зачастую информационный запрос полностью удовлетворяется ответом на вопросы «кто», «что», «где», «когда»; такой ответ вполне может быть обеспечен поверхностной семантической интерпретацией текста.

Как правило, современные реализации механизма SRL опираются на машинное обучение с использованием лексикографической информации о предикатах, обучающего корпуса, а также семантико-синтаксических свойств (хотя встречаются и системы, основанные на правилах). В то время как для английского языка все три составляющих доступны (существуют обширные словарные ресурсы, большие корпуса и надежные синтаксические парсеры), для других языков всё обстоит не так благополучно. Вследствие недостатка необходимых ресурсов для успешной реализации SRL-алгоритмов в языках, отличных от английского, а также высокой трудоёмкости создания таких ресурсов, наблюдается смещение внимания исследователей собственно SRL в сторону

* Работа проводилась при финансовой поддержке Министерства образования и науки Российской Федерации в рамках договора № 13.G25.31.0033 от "07" сентября 2010 г.

разработки более совершенных алгоритмов для английского языка. Эти алгоритмы, несмотря на очевидный прогресс и рост качества результирующих систем, опираются на определенные свойства английского языка, которые в общем случае нехарактерны для других языков. Как правило, такие алгоритмы не могут быть перенесены на другой язык без потери качества.

В последние годы возрос интерес к автоматическим и полуавтоматическим методам выделения семантических ролей (semantic role induction), которые позволили бы сократить затраты на формирование лексических баз и обучающих выборок для SRL из неанглоязычных корпусов. Это направление работ кажется перспективным, и в результате могло бы привести к сокращению дистанции между исследованиями по автоматической разметке актантов для английского и для многих других языков.

Настоящая работа представляет собой обзор теоретической и практической составляющих задачи по автоматической разметке семантических ролей. Разделы 2 и 3 содержат базовую информацию о теории семантических ролей: описание терминологии и используемых моделей, краткий взгляд на динамику развития теории в последние десятилетия, а также обзор основных проблем, которые на сегодняшний день есть в этой области. В Разделе 4 речь идёт о практических аспектах применения теории семантических ролей в автоматической обработке естественного языка: вводится дополнительный терминологический аппарат, описываются основные технологии и подходы, применяемые при автоматической разметке семантических актантов. Раздел 5 посвящён проблемам, которые стоят перед направлением SRL в настоящий момент. Раздел 6 подводит итог предлагаемого обзора и отмечает перспективные с нашей точки зрения пути развития описываемой теории и практических компонентов на её основе.

2. ТЕОРИЯ СЕМАНТИЧЕСКИХ РОЛЕЙ: ИСТОРИЧЕСКАЯ ПЕРСПЕКТИВА

Семантические роли как способ классификации и группировки участников той или иной ситуации – один из старейших конструктов в теории языка. Так, например, уже в трудах Панини (см. [1, 2]) мы можем встретить инвентарь ролей-*kāraka*, включающий в себя такие роли как «деятель», «объект», «инструмент», «реципиент», «локус» и вполне сопоставимый с инвентарями, которые предлагают современные исследователи. Однако теория семантических ролей в современном её понимании была предложена в конце 60-х гг. Ч. Филлмором [3]. Характерной чертой современной трактовки семантических ролей стала привязка к синтаксическому выражению соответствующих актантов, упор на разработку инвентаря семантических типов аргументов и классификации предикатов на основании наборов семантических ролей. Эта теория была развита в последующие десятилетия совместными усилиями синтаксистов и лексикографов (см., например, [4, 5]), в т.ч. в рамках исследований Московской лексикографической школы [6, 7] и

Модели «Смысл-текст» [8]. В результате этого развития, сопряжённого как с теоретическим, так и с практическим изучением вопроса, на сегодняшний день в распоряжении исследователя оказывается достаточно устойчивая модель с известным набором тонких и спорных аспектов, которые по-разному разрешаются в многочисленных вариантах теории.

3. СЕМАНТИЧЕСКИЕ ВАЛЕНТНОСТИ И РОЛИ

Прежде чем дать определение семантической роли и углубиться в описание собственно теории, необходимо ввести описательный аппарат семантических валентностей, которые фактически являются низкоуровневым элементом теории семантических ролей и во многом служат для них основой.

Синтаксическая структура предложения во многом определяется селективными свойствами входящих в это предложение лексем. Эти селективные свойства, по аналогии с химией, называются **валентностями**.

Если рассматривать значение лексемы с точки зрения её толкования в виде сентенциальной формы, то **семантической валентностью** (или **партиципантом**) лексемы L будет называться любая несвязанная переменная X, входящая в толкование этой лексемы [9]. Лексема, имеющая одну или более семантическую валентность, называется **предикатом** или **предикатным словом**. Набор семантических валентностей является важным элементом значения предиката, и минимальные изменения в этом наборе зачастую влекут за собой серьёзное искажение семантики. В качестве иллюстрации можно обратиться к классическому примеру описания значения слова *аренда* [6]: А арендует С значит, в первом приближении, что за какое-то вознаграждение D лицо А приобретает у лица В право на эксплуатацию недвижимой собственности С в течение времени T. В этом описании ситуации аренды представлены пять валентностей, которые можно условно обозначить как субъект аренды, объект аренды, контрагент, плата и срок. Эти валентности «достаточны и необходимы» для описания значения, и любое изменение их состава влечет за собой изменение смысла. Так, например, если удалить из толкования валентность *срок*, мы получаем ситуацию продажи, а удалив валентность *плата* – ситуацию займа. Под **семантическим актантом** понимается конкретная реализация семантической валентности в предложении.

Одна из важнейших лингвистических закономерностей, лежащих, в том числе, в основе теории семантических ролей, заключается в том, что существует согласованность между семантической и синтаксической сочетаемостью лексем, т.е. в конкретной реализации того или иного предиката каждому семантическому актанту с большой вероятностью будет соответствовать **синтаксический актант**. Это обстоятельство сделало возможным создание компактных и ёмких представлений для описания значения и сочетаемости предикатных слов. Примером такого представления может служить **модель управления** [6, 8], которая в табличном виде

указывает способ отображения семантических валентностей из толкования лексемы на синтаксический уровень. В качестве иллюстрации обратимся к определению значения лексемы *курить* из Толково-комбинаторного словаря [7]:

Х курит Y = X вдыхает через рот непосредственно из специального устройства Y дым тлеющего в Y-е вещества Z, обычно – табака, и вдыхает этот дым – с целью, чтобы это каузировало то, что X испытывает приятное ощущение.

1 = X [кто вдыхает]	2 = Y [из какого устройства вдыхает]	3 = Z [какое вещество тлеет]
S _{им}	S _{вин}	---

Витя курит трубку <сигареты, сигару, «Казбек»>

Обратим внимание на то, что третья семантическая валентность в данном варианте употребления глагола *курить* не реализуется на синтаксическом уровне (о чём свидетельствует прочерк в соответствующей графе). Существует, однако, другой вариант употребления, в котором эта валентность реализуется:

1 = X [кто вдыхает]	2 = Z [какое вещество тлеет]	3 = Y [из какого устройства вдыхает]
S _{им}	S _{вин}	из S _{род}

Витя курит «Золотое руно» из пеньковой трубки

В целом отклонения от принципа соответствия семантических и синтаксических ограничений достаточно хорошо изучены и объясняются ограниченным набором лингвистических явлений, описание и подробный разбор которых выходят за рамки данной работы. В качестве примера подобных явлений можно привести

- Нереализуемые валентности, т.е. партиципенты, которые при данном предикате обычно не выражаются, например, **Он промахнулся по зайцу пулей* [8].
- Инкорпорированный или фиксированный актант [10-13], когда данные о конкретной реализации актанта заложены в значение слова, например, *подковать, покрасить* (ср. *набить подковы, покрыть краской*).

- Явления контекстного характера (когда значение актанта ясно из контекста и потому опускается).

Более подробное и полное описание этих явлений приводится, например, в работе [14].

Целесообразность выделения **семантических ролей** как отдельного конструкта обусловлена, с одной стороны, семантическим сходством валентностей у различных предикатов, с другой стороны, наличием корреляций между сгруппированными по этому сходству валентностями и их синтаксической реализацией.

Так, например, актанты, выражаемые именными группами в именительном и творительном падежах в конструкциях с глаголами *бить, рубить, колоть, резать*, принадлежат к двум группам, члены каждой из

которых выполняют одинаковую роль по отношению к действию и, кроме того, имеют тенденцию к единообразному морфологическому и синтаксическому кодированию в предложении.

На основании этих наблюдений было предложено множество наборов семантических ролей, различающихся с точки зрения критериев объединения валентностей в ролевые группы. В начальном варианте теории семантическим ролям приписывалось два свойства:

- Семантические роли неделимы.
- У одного предиката не может быть двух актантов с одинаковой ролью.

В результате того, что, как выяснилось в ходе последовавших исследований, ни первое, ни второе свойство не соблюдаются строго на всём пространстве возможных языковых ситуаций, были предложены альтернативные модели и критерии разграничения.

Так, например, в [15] и [4] предлагается использовать инвентарь, состоящий только из двух макроролей (Прото-агенса и Прото-пациенса) с целью уменьшения **эффекта фрагментации ролей**, когда в результате синтаксического и семантического анализа исходные роли разделяются на более мелкие. Другой пример компактной ролевой системы можно видеть в работе Т. Рейнхарт [16], где предлагается для классификации ролей использовать два бинарных признака – причинность (*cause*+/-) и ментальность (*mental*+/-) аргумента. При этом допускается нулевое значение одного из признаков, что в результате предоставляет в распоряжение лингвиста более гибкую систему. Так, например, Агенса в системе Рейнхарт определяется как [+c][+m], Инструмент имеет признаки [+c][-m], при этом Причина обладает лишь признаком [+c], и таким образом совместима и с Агенсиом, и с Инструментом (см. также [17]).

На другом конце спектра находятся модели, в которых объединение валентностей в семантические роли подчиняется строгим критерием семантического сходства. В результате этого возникают инвентари, в которых подобное объединение оказывается скорее исключением, нежели правилом. Частично такой подход реализован в системе FrameNet, однако там дробность деления компенсируется механизмом семантических типов (*semantic type*), который позволяет объединять конкретные валентности в группы, а также механизмом отношений между фреймами (подробнее см. [18]). Кроме того, в большом количестве такие инвентари разрабатываются при решении конкретных прикладных задач, например, при создании узкоспециализированных систем автоматического извлечения информации.

Основная часть существующих на данный момент инвентарей в плане числа ролей находится между этими двумя подходами и содержит порядка 10 элементов, большинство из которых совпадают с предложенными Ч. Филлмором в [19]. Как правило, в список включаются следующие роли:

- Агенси – одушевленный инициатор события, способный по своей воле его прекратить.

- **Пациенс** – партиципant, наиболее вовлеченный в событие и претерпевающий наиболее значительные изменения.
- **Бенефактив** – участник, чьи интересы наиболее затронуты в ходе ситуации.
- **Экспериенцер** – получатель информации при глаголах восприятия.
- **Стимул** – источник информации при глаголах восприятия.
- **Инструмент** – неодушевленный объект, с помощью которого осуществляется действие, но который при этом не претерпевает изменений.
- **Адресат** – получатель сообщения при глаголах речи.
- **Источник** – место, из которого осуществляется движение.
- **Цель** – место, в которое осуществляется движение.

Этот список может дополняться и сокращаться в зависимости от критериев разделения и слияния ролей. Так, например, при усилении критерия единообразия грамматического отображения можно было бы объединить в одну роль Бенефактив и Адресата (т.к. они оба чаще всего маркируются дативом) или добавить роль Маршрут (т.к. она маркируется специфическими предложными группами).

Вопрос о том, какого рода соответствие существует между семантическими ролями и их морфосинтаксическим выражением, тесно связан с вопросом определения инвентаря и является не менее существенным для работоспособности теории в целом. Корреляция между семантическими ролями и их конкретными синтаксическими реализациями наблюдается достаточно широко, однако не строится по принципу однозначного соответствия и потому нуждается в моделировании. В рамках этого направления исследований можно выделить два подхода:

- Представление в словарной статье синтаксической информации наряду с семантической (например, Модели управления [6, 8] и субкатегориальные рамки [5]), т.е., фактически, **отказ от динамического моделирования зависимости в пользу словаря**.
- Представление в словарной статье только семантической информации, плюс регулярные **правила отображения** валентностей предиката на синтаксические актанты.

Подходы, основанные на правилах отображения, можно в свою очередь условно разделить на два типа: **правила с использованием иерархий** и **правила прямого отображения**. Теории, опирающиеся на правила с использованием иерархий, постулируют возможность упорядочить семантические роли в соответствии с приоритетом при заполнении позиций в синтаксической структуре [20, 21] и др. Существование подобной иерархии позволило бы создать весьма компактное и мощное описание механизма отображения семантических валентностей, однако на сегодняшний день успешной и универсальной иерархии семантических ролей создать так и не удалось (см. [23]). Правила прямого отображения задаются для каждой семантической роли вне зависимости от остальных аргу-

ментов. Фактически такой подход является попыткой моделирования структур, используемых в словарных представлениях. Задача автоматической разметки актантов, о которой пойдет речь в следующем разделе, также может быть интерпретирована как задача моделирования и вывода правил отображения в обращенном варианте: из пространства синтаксических актантов в пространство семантических.

Для полноты обзора необходимо также упомянуть о таком понятии как **ролевые типы глаголов**. После того, как инвентарь семантических ролей зафиксирован, предикаты могут быть объединены на основании совпадения или близости ролевых наборов. Предикаты, входящие в одну группу, имеют определенные общие свойства семантического характера. Множество таких ролевых групп образует **классификацию глаголов по ролевым типам**. Подобные классификации предикатов не играют первостепенной роли в современных технологиях разметки актантов, поэтому в данной работе они не рассматриваются. Подробнее см., например, [24, 9].

4. СЕМАНТИЧЕСКИЕ РОЛИ И АВТОМАТИЧЕСКАЯ ОБРАБОТКА ЯЗЫКА

Последнее десятилетие характеризуется ростом внимания к концепции семантических ролей со стороны исследователей автоматической обработки естественного языка. Основное направление деятельности в этой связи связано с автоматической разметкой актантов (semantic role labeling). Автоматическая разметка актантов как задача состоит в **определении аргументов предиката в тексте и приписывании этим аргументам семантических меток** [25].

На эту тему ежегодно появляется множество работ, регулярно проводятся крупные соревнования систем (например, Senseval-3 в 2004-м году, CoNLL в 2004 и 2005-м). На сегодняшний день для английского языка уже достигнуты приемлемые результаты, которые позволяют использовать технологию в промышленных системах.

Одна из причин, по которым semantic role labeling на сегодняшний день принадлежит к числу приоритетных направлений в обработке естественного языка, состоит в том, что аппарат семантических ролей предоставляет уровень семантической интерпретации, достаточный для решения множества прикладных задач. Так, автоматическая разметка актантов зачастую позволяет ответить на классический набор вопросов «кто», «что», «где», «когда», «почему» что покрывает большую часть информационных потребностей пользователя в таких приложениях, как вопросно-ответные системы [26] и извлечение информации [27]. Кроме того, информация, предоставляемая на данном уровне, успешно используется для снятия омонимии [28], при решении задач машинного перевода [29] и может быть использована почти в любой задаче по автоматической обработке языка, где требуется обращаться к семантической интерпретации текста.

Кроме того, концепция семантических ролей позволяет создавать значительно более адекватные и компактные языковые модели, что в результате приводит к упрощению разработки подобных систем за счёт добавления стройного и хорошо мотивированного семантического уровня абстракции над синтаксическим уровнем.

Было бы ошибочным утверждать, что автоматическое выделение актантов появилось в начале 2000-х гг. и ранее никем не использовалось, так как уже тогда существовало множество систем, основанных на правилах и шаблонах, которые были в состоянии решать подобные задачи. Основной проблемой таких систем была узкая специализация и высокие затраты на разработку, так как соответствующие правила привязки составлялись вручную для небольших групп предикатов в рамках узких тематических областей (примеры современных систем, основанных на правилах, см. например, в [30] или [31]). Машинное обучение, которое сегодня является доминирующей парадигмой в автоматической разметке актантов, было невозможно до тех пор, пока в распоряжении ученых не оказались большие массивы аннотированных данных: FrameNet [32], PropBank [33], NomBank [34] и другие.

Работа Д. Журафски и Д. Гилдея [35] во многом определила путь, по которому стало развиваться рассматриваемое направление. На сегодняшний день абсолютное большинство предлагаемых подходов трактуют поиск актантов как задачу классификации, которую можно решить с помощью соответствующих методов машинного обучения [36, 37]. Подходы, основанные на машинном обучении, в перспективе дают возможность решения, которое было бы универсальным и не зависящим от языка и от конкретной области знаний. На практике эти подходы также сталкиваются с набором трудностей, как специфических для машинного обучения, так и связанных с внешними причинами ресурсного характера. Далее мы рассмотрим структуру типичной системы для автоматической разметки актантов, опишем основные технологии, применяемые при решении задачи, а также приведем краткий обзор проблем, характерных для машинного обучения в целом и для задачи распознавания актантов в частности.

За небольшим исключением поиск и семантическая интерпретация актантов в системах автоматической разметки ролей опираются на некоторую синтаксическую репрезентацию исходного текста [38, 39]. Поскольку для успешного применения машинного обучения требуются большие объемы данных, как правило, речь в данном случае идёт о корпусе текстов, содержащем метки ролей поверх синтаксической разметки.

Задача автоматического распознавания актантов в общем случае состоит из трех подзадач:

- Обнаружение в тексте целевых предикатов, т.е. таких слов, которые инициируют ту или иную ситуацию или фрейм.
- Обнаружение аргументов предиката.
- Классификация аргументов.

Несмотря на распространённую точку зрения о том, что лучшее качество работы системы достигается при одновременном решении всех трёх подзадач, с точки зрения архитектуры, контроля над системой и вычислительной эффективности, предпочтение отдаётся последовательному и независимому решению этих подзадач.

Задача **по обнаружению в тексте целевых предикатов** решается на уровне морфологического анализа и разрешения неоднозначностей, т.е. без использования семантических ролей (хотя, как упоминалось выше, в дальнейшем механизм автоматической разметки актантов может быть применен для дополнительной фильтрации неоднозначных случаев).

Обнаружение аргументов трактуется как задача бинарной классификации, в которой на основании входных данных, свойств и модели для каждого синтаксического узла необходимо определить, является он аргументом данного предиката или нет. По крайней мере для английского языка имеются свидетельства в пользу того, что в задаче обнаружения аргументов следует использовать иной набор свойств, чем при дальнейшем распределении актантов по ролям [37]. Кроме того, для поиска аргументов в английском языке достигнуто достаточно высокое качество работы соответствующих модулей, так что изоляция этого шага выглядит оптимальным и верным решением [40]. Общепринятой практикой при реализации данного шага является максимизация полноты (т.е. выдача максимального числа вариантов интерпретации) с дальнейшим отсеком нерелевантных аргументов на этапе классификации аргументов по ролям [41].

Классификатор аргументов получает на вход набор узлов синтаксической структуры, которые были определены как аргументы анализируемого предиката, и на основании набора свойств и модели **приписывает аргументу ту или иную семантическую роль**.

Оставив задачу по поиску предиката в тексте за рамками обсуждения по причине её явной вторичности и неспецифичности¹, остановимся немного подробнее на технологиях классификации и конкретных алгоритмах, которые применяются для решения оставшихся двух подзадач.

В общем случае **задача классификации** формулируется следующим образом. Пусть $A = \{a_1, a_2, a_3, \dots\}$ - множество **объектов**, а $C = \{c_1, c_2, c_3\}$ - множество наименований **классов** (или категорий), по которым объекты могут быть распределены. Существует некоторая неизвестная **целевая функция** $\Phi = C \times A$, принимающая значение 0 или 1 в зависимости от того, принадлежит ли объект классу. Для части объектов класс известен, они образуют множество, называемое **обучающей выборкой**: $A^m = \{(a_1, c_1), (a_2, c_2), \dots, (a_m, c_m)\}$. Задача состоит в том, чтобы на основании обучающей выборки построить **алгоритм**, который позволял бы определить класс для произвольного объекта $a \in A$ и результаты работы которого приближались бы к значениям неизвестной целевой функции Φ .

¹ Поиск целевого предиката в тексте с точки зрения автоматической обработки языка мало чем отличается от поиска любого другого слова

В рамках такого подхода объекты описываются в виде **набора признаков**. Фактически, признак представляет собой отображение $f: A \rightarrow Df$, где Df – область значений признака. При таком обозначении каждый конкретный объект $a \in A$ может быть описан в виде **вектора значений признаков** ($f_1(a), f_2(a), f_3(a) \dots f_n(a)$). Множество признаков со всеми возможными значениями в совокупности образует **признаковое пространство**.

Работа классификатора проходит в два этапа. На **этапе обучения** классификатор получает на вход множество признаков векторов, соответствующих объектам обучающей выборки, с метками классов, которые присвоены этим объектам, и строит **модель данных**. В соответствии с этой моделью на **этапе применения** классификатор, получив на вход признаковое описание объекта, присваивает этому объекту метку класса.

5. ПРОБЛЕМЫ АВТОМАТИЧЕСКОЙ РАЗМЕТКИ АКТАНТОВ

Качество работы классификатора (т.е. степень соответствия присваиваемых классов реальным значениям) определяется тремя факторами: размером и качеством обучающей выборки, набором признаков и собственно алгоритмом классификации. Недостаточно большая, неоднородная или наоборот, слишком однородная выборка, нерелевантный набор признаков или (в меньшей степени) неудачный выбор алгоритма способны серьёзным образом повлиять на качество модели и, следовательно, на качество работы соответствующих лингвистических модулей.

В отличие от низкоуровневых классификационных задач по автоматической обработке текста (например, снятия частеречной омонимии), задача выделения семантических актантов характеризуется **крайне большим признаковым пространством**: наборы свойств, используемые при выделении семантических актантов, как правило, включают в себя синтаксическую и семантическую информацию [25]. Так, например, в [39] предлагается использовать для классификации каждого аргумента такие свойства, как:

- набор лемм зависимых слов у целевого предиката;
- лемму родительской вершины предиката;
- набор предполагаемых валентностей предиката (из словаря);
- часть речи предиката;
- залог предиката;
- лемму и часть речи аргумента;
- путь по синтаксической структуре от вершины к аргументу и т.д.

Следует отметить, что одно только свойство «лемма» может принимать сотни тысяч возможных значений (описываемый набор признаков был предложен для английского языка), при этом таких свойств используется несколько, и эти свойства формируются для каждого объекта в выборке. Уменьшение размерности и снижение сложности задачи классификации достигается, как правило, за счёт **разбиения на подзадачи** и **подбора признаков**, которые имели бы высокую различительную силу и по-

зволяли бы наилучшим образом смоделировать ситуацию, подлежащую анализу.

Так, например, описанный выше трехступенчатый способ организации лингвистического процессора позволяет значительно упростить задачу классификации аргументов за счёт предварительного отсева синтаксических узлов, которые аргументами не являются.

В качестве иллюстрации уменьшения размерности за счёт подбора признаков можно привести использование результатов предварительной морфологической и синтаксической обработки, а также словарной информации вместо данных символического уровня (например, лемм). Опора на **лингвистические данные высоких уровней**, тем не менее, требует наличия для исследуемого языка либо **обширных корпусов** со всей необходимой разметкой, либо систем, которые позволяли бы выполнять такую разметку **автоматически** (если речь идёт о промышленных системах, второе требование строго обязательно, так как на вход подаётся в лучшем случае чистый текст). При отсутствии таких ресурсов для конкретного языка активная разработка технологии автоматического выделения актантов для него также становится трудноосуществимой.

Под влиянием общей тенденции, в последние годы наблюдается значительный рост активности по созданию корпусных ресурсов, ориентированных на хранение информации о семантико-синтаксической сочетаемости слов. Вслед за многочисленными ресурсами, ориентированными на английский язык, были созданы Prague Dependency Treebank [42] для чешского языка, Salsa Project [43] – для немецкого языка, Chinese PropBank [44] и Chinese NomBank [45] – для китайского языка. Также активно разрабатываются версии FrameNet для немецкого, испанского, японского и шведского языков, в работе находится и FrameNet-ориентированный ресурс для русского языка [46]. Хотя зачастую подобные ресурсы позиционируются в первую очередь как перспективные лексикографические системы, реализующие гибридный принцип (сочетание словаря и корпуса), создание подобных аннотированных корпусов в перспективе позволит использовать их и для решения задач автоматической обработки текста. Существенное влияние на скорость разработки этих ресурсов оказывает доступность качественных средств автоматической разметки текстов, начиная от токенизаторов (ресурсов для разбиения текста на слова и пунктуацию) и заканчивая синтаксическим и семантическим анализом.

В контексте подобных инструментов поддержки кажется важным остановиться на сравнительно новом направлении исследований, в большой степени связанном с описываемой в данной работе задачей. Речь идёт о разработке технологий по **извлечению семантических ролей для произвольного предиката**, или semantic role induction. Мотивация для исследований во многом соответствует тому, о чём говорилось выше: для успешной работы систем автоматической разметки актантов необходимы большие объёмы обучающих данных, создание которых вручную, в свою очередь, требуют больших трудозатрат [47]. Один из базовых методов автома-

тической генерации фреймов основывается на трехступенчатой схеме, используемой при выделении актантов и подробно описанной выше. Отличие состоит в том, что на третьем этапе вместо задачи классификации (т.е. приписывания аргументу ролевой метки) решается задача кластеризации: для каждого предиката аргументы, обнаруженные в корпусе, группируются в соответствии с некоторой мерой синтаксического сходства. Таким образом, формируется набор кластеров, каждый из которых соответствует реализации той или иной семантической роли при данном предикате. Такой подход в целом хорошо согласуется с теоретической моделью, лежащей в основе семантических ролей, и в перспективе должен позволить автоматически или полуавтоматически пополнять семантико-синтаксические словари, что значительно ускорит разработку подобных ресурсов для языков, для которых они ещё в слабой степени разработаны. Подробнее см., например, [48].

В завершение кажется важным обратиться к проблеме межязыковой переносимости технологий обработки естественного языка, которая актуальна для задачи автоматической разметки актантов не в меньшей степени, чем для многих других задач, особенно принимая во внимание уже неоднократно упомянутый «англоцентризм» современных разработок в этой области. Наличие необходимых низкоуровневых ресурсов и обучающих корпусов само по себе не гарантирует применимость разрабатываемых сегодня алгоритмов автоматического выделения актантов, так как большинство алгоритмов разрабатываются для английского языка и являются в большой степени языкозависимыми. Успех переноса алгоритма с одного языка на другой в большой степени зависит от структурного сходства этих языков в части используемых в алгоритмах свойств. Так, например, алгоритмы, разработанные для английского языка, неоднократно успешно портировались на китайский язык [25, 49] с почти полным сохранением набора свойств, в том числе и позиционных. Прямой перенос этих алгоритмов на язык со свободным порядком слов, например, на русский, привел бы к серьёзным нарушениям работы алгоритмов: позиционное свойство, обладающее достаточно большой различительной силой в английском языке, было бы в значительной степени ослаблено, и для того, чтобы компенсировать это, нужно было бы подключить морфологические признаки. Описанная проблема имеет системный характер и относится не только к задаче автоматического выделения актантов. Сложности с переносом алгоритмов, доказавших свою состоятельность на английском материале, возникают, например, в задачах снятия морфологической неоднозначности или поиска имен собственных в неструктурированных текстах. Хорошая переносимость моделей с одного языка на другой, возможно, могла бы быть достигнута в рамках исследований по сопоставлению статистических и типологических характеристик языков на основе обширного корпусного материала.

6. ЗАКЛЮЧЕНИЕ

Итак, мы вкратце описали теоретический и практический аспекты автоматического выделения актантов в кросс-лингвистической перспективе.

Концепция семантических ролей, на которой базируется данная технология, на сегодняшний день поддерживается уже в значительной степени сформировавшейся теорией, которая, несмотря на то, что согласие по некоторым проблемам до сих пор не достигнуто, предоставляет лингвисту достаточно прочную и проверенную в ходе многочисленных исследований платформу. Сегодня существует множество интерпретаций концепции семантических ролей, которые отталкиваются от общей базы, заложенной в 70-е гг. XX в. при участии таких лингвистов как Дж. Грубер [50], Ю.Д. Апресян [51] и Ч. Филлмор [20]. Различия в современных интерпретациях касаются в основном принципов построения инвентаря семантических ролей, а также способа представления информации о синтаксической сочетаемости (в словаре или с помощью правил). По всей видимости, чертами современного подхода следует признать тенденцию к дроблению ролей (см., например, [18] и [46]) и кодированию информации о синтаксической сочетаемости в словаре (а в случае с гибридными лексикографическими системами также и в корпусе). Поскольку крупные корпусные ресурсы, в которых представлена информация о распределении актантов по предикатам, отказываются от стандартизированных инвентарей в пользу частных ролей, современные методы автоматического выделения актантов, которые ориентируются на эти ресурсы, также результатом своей работы имеют распределение аргументов каждого предиката по достаточно частным категориям. Это иллюстрирует важную функцию семантических ролей как модели, определяющей и формулирующей задачу для автоматических систем на её основе: автоматическая система будет стремиться к разным результатам в зависимости от того, поведение какой модели она пытается воспроизвести.

В отличие от теории семантических ролей, которая, как уже было сказано выше, на сегодняшний день пребывает в более-менее стабильном состоянии, практическая область, связанная с автоматическим выделением актантов, активно развивается. Основные причины привлекательности этого направления состоят в том, что аппарат семантических ролей позволяет создать промежуточный уровень семантической интерпретации текста, которого, несмотря на неполноту, зачастую оказывается достаточно для решения прикладных задач. Система автоматической разметки актантов может быть полезна не только при целевом применении (выделение аргументов предиката с последующим заполнением фреймов), но и в качестве источника вспомогательной информации для задач разрешения омонимии, машинного перевода и прочих. Рост популярности этого направления исследований в последние годы не в последнюю очередь обусловлен тем, что появились многочисленные корпусные ресурсы, предоставляющие значительные объёмы текстовой информации, аннотированной в том числе в терминах семантических ролей пред-

ставленных в текстах актантов. Работа [35] продемонстрировала перспективность использования методов машинного обучения для решения задачи автоматической разметки актантов и стала одним из источников того активного развития, которое мы можем наблюдать сегодня в этой области.

Наличием больших объемов аннотированных данных объясняется и высокая степень смещения фокуса современных исследований в сторону английского языка, т.к. большинство подобных ресурсов создано именно для английского. Кроме того, английский язык является привлекательным для исследователей в этой области за счёт большого числа ресурсов для синтаксической и словарной предобработки, находящихся в открытом доступе.

В то же время, подобный «англоцентризм» влечет за собой несколько существенных проблем для других языков. Значительные силы сейчас сконцентрированы на разработке алгоритмов автоматического выделения актантов для английского языка. Другие языки при этом разрабатываются в значительно меньшей степени, и успех подобных разработок главным образом зависит от наличия крупных корпусных ресурсов, качественных синтаксических парсеров и своего рода *типологической «совместимости»* этих языков с английским. Так, например, свойства и модели, используемые при построении систем на основе машинного обучения для английского, сравнительно легко портируются на китайский язык. Одной из главных характеристик систем лингвистической обработки текста, ориентированных на английский язык, является акцент на использование позиционных свойств, которые, строго говоря, не всегда оказываются сильным различающим элементом для решения прикладных лингвистических задач методами машинного обучения. Тема типологической совместимости языков в свете машинного обучения на данный момент изучена достаточно мало и нуждается в дальнейшей проработке. Разработка аппарата для обеспечения такой совместимости позволила бы значительно упростить разработку систем, основанных на машинном обучении и ориентированных на анализ естественного языка.

Помимо этого, перспективными в данном отношении кажутся технологии с применением обучения без учителя для поддержки наполнения семантико-синтаксических словарей. Такие технологии могли бы значительно ускорить создание соответствующих ресурсов для языков, до сих пор не в полной мере ими обеспеченных. Они позволили бы сократить дистанцию в области автоматического выделения актантов между английским языком, где это задача уже почти решена и где технология используется в промышленных системах, и другими языками, в которых к решению этой задачи только начинают приступать.

СПИСОК ЛИТЕРАТУРЫ

- Misra V. The Descriptive Technique of Panini. – Mouton: The Hague, 1966.
- Dahiya Y. Panini as a linguist: Ideas and Patterns. – Dehli: Eastern Book Linkers, 1995.
- Fillmore C.J. Some problems for case grammar // 22nd annual Round Table. Linguistics: developments of the sixties – viewpoints of the seventies, volume 24 of Monograph Series on Language and Linguistics. – Washington: Georgetown University Press, 1971.
- Dowty D. Thematic proto-roles and argument selection // Language, – 1991. – Vol. 67(3).
- Jackendoff R. Semantic Interpretation in Generative Grammar. – Cambridge: MIT Press, 1972.
- Апресян Ю.Д. Лексическая семантика: Синонимические средства языка // Избранные труды, т. I. – М.: ЯПК. 1995 (2-е изд.), 1974.
- Толково-комбинаторный словарь русского языка // Wiener Slawistischer Almanach. Sonderband 14. – 1984.
- Мельчук И.А. Опыт теории лингвистических моделей "Смысл-Текст". – М.: Наука, 1974.
- Тестелец Я.Г. Введение в общий синтаксис. – М.: РГГУ, 2001.
- Бирюлин Л.А. Диатезы русских глаголов, обозначающих атмосферные явления. – Л.: АКД, 1984.
- Кацнельсон С. Д. К понятию типов валентности // Вопросы языкознания. – 1987. – № 3.
- Jackendoff R. Semantic Structures. – Cambridge: MIT Press, 1990.
- Рахилина Е.В. Семантика или синтаксис? К анализу частных вопросов в русском языке. – München: Sagner, 1990.
- Апресян Ю.Д. Типы соответствия семантических и синтаксических актантов // Проблемы типологии и общей лингвистики. Материалы. – 2006.
- Van Valin R. D. A synopsis of Role and Reference Grammar // Advances in Role and Reference Grammar. – Amsterdam: Benjamins, 1993.
- Reinhart T. The theta system: An overview // Theoretical Linguistics. – 2002. – Vol. 28.
- Dimitriadis A. An Event Semantics for the Theta System. – Utrecht: Utrecht institute of Linguistics, 2004.
- Fillmore C.J. Border conflicts: FrameNet meets Construction Grammar // Lecture at EURALEX XIII. – 2008.
- Fillmore C.J. The Grammar of Hitting and Breaking // Readings in English Transformational Grammar. – Washington: Georgetown University Press, 1970.
- Fillmore C.J. The Case for Case // Universals in Linguistic Theory. – New York: Holt, Rinehart & Winston, 1968.
- Larson R.K. Implicit arguments in Situation Semantics // Linguistics and Philosophy. – 1988. – Vol. 11.
- Lazard G. L'actance. – Paris: Presses Universitaires de France, 1994.
- Падучева Е.В. Структура события: семантические роли, аспектуальность, каузация // НТИ. Сер. 2. – 2009. – № 6.
- Яхонтов С.Е. Классы глаголов и падежное оформление актантов // Проблемы теории грамматического залога. – Л.: Наука, 1978.

25. Xue N. Labeling Chinese predicates with semantic roles // *Computational Linguistics*— 2008. – Vol. 24, N 2.
26. Narayanan S., Harabagiu S. Question answering based on semantic structures. // *Proceedings of the 20th International Conference on Computational Linguistics*. – 2004.
27. Surdeanu M., Harabagiu S., Williams J., Aarseth P. Using predicate-argument structures for information extraction // *Proceedings of ACL-2003*. – 2003. – Vol.1.
28. Кустова Г.И., Толдова С.Ю. НКРЯ: семантические фильтры для разрешения многозначности глаголов // *Национальный корпус русского языка: 2006—2008. Новые результаты и перспективы*. – СПб: Нестор-история. – 2009.
29. Boas H. Bilingual FrameNet dictionaries for machine translation. // *Proceedings of LREC-2002*, 2002.
30. Ермаков А.Е., Плешко В.В. Семантическая интерпретация в системах компьютерного анализа текста // *Информационные технологии*. – 2009. – № 6.
31. Bick E., Ibanez M. Automatic Semantic Role Annotation for Spanish. // *Proceedings of NODALIDA-2009*. – 2009.
32. Baker C. F., Fillmore C. J., Lowe J.B. The Berkeley FrameNet Project. // *Proceedings of COLING-ACL-98*. – 1998.
33. Palmer M., Gildea D., Kingsbury P. The Proposition Bank: A Corpus Annotated with Semantic Roles // *Computational Linguistics*— 2005. – Vol. 31, N 1.
34. Meyers A., Reeves R., Macleod C., Szekely R., Zielinska V., Young B., Grishman R. The NomBank Project: An interim report // *Proceedings of the NAACL-HLT Workshop on Frontiers in Corpus Annotation*. – 2004.
35. Gildea D., Jurafsky D. Automatic Labeling of Semantic Roles // *Computational Linguistics*— 2002. – Vol. 28, N 3.
36. Pradhan S., Sun H., Ward W., Martin J. H., Jurafsky D. Parsing arguments of nominalizations in English and Chinese // *Proceedings of NAACL-HLT-2004*. – 2004.
37. Xue N., Palmer M. Automatic semantic role labeling for Chinese verbs // *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. – 2005.
38. Punyakanok V., Roth D., Yih W. The necessity of syntactic parsing for semantic role labeling // *Proceedings of IJCAI-2005*. – 2005.
39. Johansson R., Nugues P. Comparing Dependency and Constituent Syntax for Frame-semantic Analysis // *Proceedings of LREC-2008*. – 2008.
40. Hacioglu K., Pradhan S., Ward W., Martin J. H., Jurafsky D. Semantic role labeling by tagging syntactic chunks // *Proceedings of CoNLL-2004*. – 2004.
41. Pradhan S., Hacioglu K., Ward W., Martin J. H., Jurafsky D. Semantic role parsing: Adding semantic structure to unstructured text // *Proceedings of ICDM-2003*. – 2003.
42. Sgall P., Panevova J., Hajicova E. Deepsyntactic annotation: Tectogrammatical representation and beyond // *Proceedings of the HLT-NAACL-2004 Workshop: Frontiers in Corpus Annotation*. – 2004.
43. Burchardt A., Erk K., Frank A., Kowalski A., Pado S., Pinkal M. The SALSA Corpus: a German corpus resource for lexical semantics // *Proceedings of LREC-2006*. – 2006.
44. Xue N., Palmer M. Annotating the propositions in the Penn Chinese Treebank // *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*. – 2003.
45. Xue N. Annotating the predicate-argument structure of Chinese nominalizations // *Proceedings of the Fifth International Conference on Language Resources and Evaluation*. – 2006.
46. Ляшевская О.Н., Кузнецова Ю.Л. Русский фреймнет: к задаче создания корпусного словаря конструкций // *Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог 2009»*. – 2009. – Вып. 8 (15).
47. Pradhan S., Ward W., Martin J. Towards robust semantic role labeling // *Computational Linguistics*— 2008. – Vol. 34, N 2.
48. Grenager T., Manning C. Unsupervised discovery of a statistical verb lexicon // *Proceedings of EMNLP-2006*. – 2006.
49. Sun H., Jurafsky D. Shallow semantic parsing of Chinese. // *Proceedings of NAACL-2004*. – 2004.
50. Gruber J.S. *Studies in Lexical Relations*. – Cambridge: MIT Press, 1965.
51. Апресян Ю.Д. Об экспериментальном толковом словаре русского языка. // *Вопросы языкознания*. – 1968. – № 5.

Материал поступил в редакцию 17.10.12.

Сведения об авторе

КУЗНЕЦОВ Илья Олегович - аспирант факультета филологии Национального исследовательского университета – Высшая школа экономики (НИУ ВШЭ), младший научный сотрудник Центра семантических технологий НИУ ВШЭ, Москва
E-mail: iokuznetsov@gmail.com

Указатель статей, опубликованных в сборнике «Научно-техническая информация», и Авторский указатель за 2012 год

Указатель статей*

ОБЩИЙ РАЗДЕЛ

Семенюк Э.П. Роль информатики в экологизации общества	1 (1) 1
Урсул А.Д. Информация и информационный подход: от информатики к глобалистике	2 (1) 1
Белкин С.В. Государственная аналитика инновационного развития в России: содержание, тенденции и актуальные задачи	3 (1) 1
Опарина О.Д. Междисциплинарные аспекты взаимодействия библиографии и социально-гуманитарных наук	3 (1) 8
Тихонов М.Н., Богословский М.М. Подводные камни новых информационно-коммуникационных технологий	4 (1) 1
Волченков Е.Я. К онтологии антропогенных систем (информационные аспекты)	4 (1) 9
Алошги Х. Р. Философский взгляд на информацию и информационную технологию	4 (2) 1
Акопян Д.А., Еляков А.Д. Проблема качества социальной информации при принятии решений	5 (1) 1
Мыльников Л.А. Системный взгляд на проблему моделирования и управления производственными инновациями	5 (1) 11
Сюнтюренко О.В. Социальные и экономические риски развития информационных технологий	6 (1) 1
Лопатина Н.В. Библиотекведение и социология: к вопросу о междисциплинарной методологической коммуникации	6 (1) 6
Плешкевич Е. А. Коммуникационная теория документа: pro et contra	7 (1) 1
Стародубов В.И., Куракова Н.Г., Цветкова Л.А., Арефьев П.Г., Кураков Ф.А. Проблемы оценки мирового уровня конкурентоспособности российской науки на примере национальной клинической медицины	8 (1) 1
Двоеносова Г. А. Признаки документа	9 (1) 1
Аншаков О. М. ДСМ-метод: теоретико-множественное объяснение	9 (2) 1

Каменская М.А. Понятие «информация» в контексте молекулярно-клеточной биологии	11 (1) 4
Борщев В.Б. Лаборатория электромоделирования ВИНТИ: 1958-1959 гг.	11 (2) 5
Двоеносова Г. А. Свойства документа	12 (1) 1
Шемакин Ю.И. Новая наука о системах (системантика) и эпистемология в теории познания	12(1) 11

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ РАБОТЫ

Громова Р.А., Романова М.С., Ставинский Е.Н., Ситникова И.С., Храмова Г.И. Информационный экспресс-анализ: полимерные нанокомпозиты	1 (1) 13
Лобанов А.С. Методика сбора и обработки информации при квалитетрическом оценивании функциональности сотрудников	1 (1) 17
Кузьмина Д.А. База данных экологических проектов РАН: пространственные характеристики исследований	2 (1) 12
Цветкова В.А., Мельникова Е.В. Инновационная направленность современной системы НТИ Франции	2 (1) 19
Брежнева В.В. Тенденции развития информационного обслуживания	3 (1) 12
Алешин Л.И. Материально-техническая база – фундамент информатизации библиотек	3 (1) 16
Лопатина Н.В., Сладкова О.Б. Информационная культура мегаполиса: единство многообразия	3 (1) 19
Арутюнов В.В. Опыт реализации специальных курсов по информационным технологиям и защите информации	3 (1) 22
Холодков К.И., Алешин И.М., Корягин В.Н., Сухорослов О.С., Шогин А.Н. Опыт развертывания грид-инфраструктуры для поддержки вычислительных веб-серверов	4 (1) 15
Али Исмоили. Стратегическое управление трудовыми ресурсами в динамической среде	4 (1) 20

* 1 означает номер сборника, (1) – серию, 1 – страницу

Алешин Л.И. Облачные библиотеки	4 (1) 26	Баяндин Н.И., Кретов В.С. Мониторинговая система ведения конкурентной разведки для предприятий малого бизнеса	12(1) 25
Московкин В.М., Шигорина Н.А., Попов Д.А. Возможности использования поискового инструмента Google Patents в патентометрическом анализе (на примере крупнейших инновационных компаний мира)	5 (1) 24	ДОКУМЕНТАЛЬНЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ	
Хорьков Д.А. Методы тестирования сетевых систем обнаружения компьютерных атак	6 (1) 9	Солошенко Н.С., Ефременкова В.М., Кириллова О.В. Анализ публикационной активности российских организаций по функциональным наноматериалам	1 (1) 24
Булдакова Т.И., Джалолов А. Ш. Анализ информационных процессов и выбор технологий обработки и защиты данных в ситуационных центрах	6 (1) 16	Пасичник Л.П. Технические условия в системе стандартизации Украины и Российской Федерации	3 (1) 26
Мулева Ю. Н. Интеграция электронных информационных ресурсов в учебную деятельность студента вуза: классическое обучение vs маркетинг	6 (1) 23	Маркусова В.А. Библиометрические показатели научных журналов для отбора в информационную систему Web of Science и другие информационные продукты компании Thomson Reuters	6 (1) 30
Арутюнов В. В. Особенности идентификации и аутентификации как основы защиты информации в компьютерных системах	7 (1) 10	Московкин В. М. Google Books и «культурологические тренды»	7 (1) 27
Мельникова Е. В. Система НТИ Германии и ее роль в развитии инновационной среды	7 (1) 16	Берёзкина Н.Ю., Сикорская О.Н., Хренова Г.С. Анализ публикационной активности ученых Национальной академии наук Беларуси	8 (1) 33
Шемберко Л.В., Боровик М.А. Информационный ресурс по социальным и гуманитарным наукам для поддержки научных исследований инновационного развития России	8 (1) 16	Ханжин А. Г., Кожокару А. А. Особенности реферативного перевода и феномен чинглиш	9 (1) 19
Нестеров А.В. О научно-квалификационных критериях	8 (1) 27	Романова М.С., Громова Р.А., Ситникова И.С. Патентный экспресс-анализ технического уровня и тенденций развития заданной области техники	10(1) 21
Арутюнов В.В. Облачные вычисления: история возникновения, современное состояние и перспективы развития	9 (1) 6	Рахимова Н.М. Статистический анализ публикационной активности научно-исследовательского института	10(1) 26
Мартirosян З. Г., Саркисян Д. Б. Сетевые программы как средство активизации учебной деятельности студентов	9 (1) 13	Буйлова Н.М., Осипов А.И. Наукометрический анализ публикаций по наноэнергетике. По материалам выпуска РЖ ВИНТИ РАН «Физика нанобъектов и нанотехнология»	11(1) 30
Шемберко Л.В., Слива А.И. Информация по философии в базах данных ИНИОН РАН	10 (1) 1	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ	
Грузова А.А. Роль службы информации организации при внедрении корпоративных информационных систем	10(1) 13	Волкова А. Ю., Шестерникова О. П. О создании интеллектуальных систем, реализующих ДСМ-метод автоматического порождения гипотез, и результатах их применения для анализа медицинских данных	5 (2) 10
Сугак Д.Б. Электронное представительство вуза в глобальной сети Интернет	10(1) 18	Лапшин В. А. Вопросно-ответные системы: развитие и перспективы	6 (2) 1
Егоров В.С. Особенности доступа к научной литературе в электронном обществе: история вопроса	11(1) 18	Виноградов Д. В. Вероятностное порождение гипотез в ДСМ-методе с помощью простейших цепей Маркова	9 (2) 20
Захарчук Т.В. Научная школа в библиографоведении: представления научного сообщества	11(1) 24	Ковтун В. А. Логико-комбинаторная система интеллектуального анализа данных, использующая аппарат нечеткого вывода	9 (2) 28
Алпатова М.Г., Антипов М.В., Леонтьева Т.М., Шогин А.Н. Новый интерфейс комплекса программных средств для обслуживания пользователей БД ВИНТИ РАН в режиме on-line	12(1) 18		

Волкова Т. А. Вариант динамического ДСМ-метода для обучения мобильного робота	9 (2) 34	Касумов В. А., Касумова К. П. Методы определения тематики и содержания индивидуальных и коллективных информационных потребностей	2 (2) 1
Финн В.К. Двенадцать тезисов об аргументационных системах	11(2) 20	Халабия М.Л. Анализ форматов библиографической записи на повторяемость элементов данных	2 (2) 10
Фабрикантова Е.Ф. Моделирование систем средствами искусственного интеллекта	12 (2) 1	Мокрозуб В. Г., Немтинов К. В., Шаронин К. А. Программное обеспечение автоматизированных систем размещения объектов в пространстве, инвариантное к предметной области	3 (2) 11
ИНФОРМАЦИОННЫЕ СИСТЕМЫ		Яцко В. А. Интерпретация закона Брэдли-Форда в терминах геометрической прогрессии	4 (2) 19
Бениаминов Е. М., Лапшин В. А. Уровни представлений онтологий, языки, математические модели и проект Веб-сервера онтологий в стиле Веб 2.0	3 (2) 1	Карпухин И. Н., Кораблин Ю. П., Незнанов А. А. Модели времени в имитационном моделировании	5 (2) 16
Марушкина А. С. Семантическое представление текста, описывающего механику силовых процессов, в универсальной онтологии	4 (2) 13	Гриняк В. М., Трофимов М. В. Оценка характера движения морских судов в лингвистических переменных	5 (2) 22
Голицына О. Л., Максимов Н. В., Окропишина О. В., Строгонов В. И. Онтологический подход к идентификации информации в задачах документального поиска	5 (2) 1	Нешиной В. В. Метод наибольшего правдоподобия, устойчивый метод и энтропия	5 (2) 27
Бургучев С.С. Организация сбора данных систем спутникового позиционирования в сети стационарных приемных пунктов	12(2) 28	Осипов А. Л., Бобров Л. К. Об использовании моделей статистического распознавания в системе виртуального скрининга химических веществ	7 (2) 1
ИНФОРМАЦИОННЫЙ ПОИСК		Петрина А. М. Наноробототехника: моделирование и эксперименты	7 (2) 7
Сергеев А. Ю., Тютюнник В. М. Методика оценки и повышения эффективности тематико-ориентированного интернет-поиска с помощью минимизации объема поисковой выборки, обеспечивающей тематическую полноту поиска	7 (2) 18	Еркимбаев А. О., Жижченко А. Б., Зицерман В. Ю., Кобзев Г. А., Сон Э. Е., Сотников А.Н. Интеграция баз данных по свойствам вещества. Подходы и технологии	8 (2) 1
ИНФОРМАЦИОННЫЕ ЯЗЫКИ		Московкин В. М. Построение кластеров результатов исследований с помощью специализированных институтов Google	8 (2) 9
Петров Т.Г., Андриянец-Буйко А.А., Мошкин С. В. Двупараметрический алфавит для кодирования структурно-химической информации и ее систематизации (на примере турмалина)	2 (2) 15	Гайдамакин Н. А., Леонтьев С. В., Ялпаев А. А. Усреднение оценок, формируемых экспертами в виде интервальных значений	8 (2) 14
Петров Т. Г. Графическое отображение процессов эволюции составов поликомпонентных объектов любой природы	3 (2) 21	Бабин М. А. О приближенном базисе импликаций	8 (2) 20
ИНФОРМАЦИОННЫЙ АНАЛИЗ		Климова С.Г., Михеенкова М.А. Формальные средства ситуационного анализа: опыт применения	10 (2) 1
Гусев В. Д., Саломатина Н. В., Кузьмин А. О., Пармон В. Н. Экспресс-анализ динамики функционирования терминов в терминологическом словаре предметной области	1 (2) 1	Яцко В.А. Методы генерации словарей	10(2) 14
Клещев А. С., Смагин С. В. Задачи индуктивного формирования знаний для онтологии медицинской диагностики	1 (2) 9	Забезжайло М.И. К проблеме быстрого анализа заголовков при адресации пакетов данных в компьютерных сетях	12 (2) 8

**Головин В.Ф., Журавлев В.В.,
Архипов М.В., Павловский В.Е.,
Орлов И.А.** Особенности математиче-
ского моделирования многосуставного
робота, взаимодействующего с упругой
динамической средой 12(2) 16

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

Логачева В. К., Клышинский Э. С.
Метод генерации конечного автомата
для задач машинной транскрипции 1 (2) 22

Лапшин В. А. Грамматический анализ
языков с неоднозначными лексикой и
синтаксисом 2 (2) 24

Архангельский Т. А. Электронные
корпуса албанского, калмыцкого, лез-
гинского и осетинского языков 4 (2) 24

Яцко В. А. Проблемы и алгоритмы
клаузуальной декомпозиции текста 6 (2) 10

Рудницкая Е. Л. Структура предложе-
ния в корейском как языке типа SOV 6 (2) 17

Соснин О. М. О соответствии русско-
английской транслитерации русскому
алфавиту 8 (2) 24

Борюшкина Е. Н. Проблемы перевода
слов *животъ, житие, жизнь* в Толко-
вой Палее 8 (2) 29

Хайруллин В. И. Информационно-
переводческое поле юридической тер-
минологии 8 (2) 33

Тур О.Н. Заимствования в современной
украинской документоведческой терми-
носистеме 10(2) 21

Федосеева Л.Н. Национально и куль-
турно обусловленная детерминирован-
ность локативной семантики в языковых
картинах мира 10(2) 27

Васильченко К.А. Информационная
специфика дефиниций 10(2) 31

**Белоногов Г.Г., Гиляревский Р.С.,
Хорошилов А.А.** Проблемы автомати-
ческой смысловой обработки текстовой
информации 11(2) 31

Кузнецов И.О. Автоматическое выде-
ление глагольных актантов: теоретиче-
ская основа и актуальные подходы 12(2) 36

СПРАВОЧНО-ИНФОРМАЦИОННЫЙ РАЗДЕЛ

Плешкевич Е.А. Эволюция институтов
информационной деятельности: обсуж-
дение выхода в свет второй части моно-
графии Б.А. Семеновкера [Рец на кн.] 1 (1) 30

Буйлова Н.М., Осипов А.И. Наукомет-
рический анализ документов, представ-
ленных на IV Международном форуме
по нанотехнологиям (Москва, 2011 г.) 2 (1) 30

Лейчик В.М. О книге А.С. Герда
[Рец. на кн.] 3 (1) 32

Силков С.В. О словаре-справочнике
терминов и понятий «Документоведен-
ие» Г.Н. Швецов-Водки [Рец на кн.] 4 (1) 29

Плешкевич Е.А. Может ли документ
быть частным случаем информации? 5 (1) 31

Плющ М.А. О доступных источниках
информации о библиотеках А.П. Во-
лынского и его сторонников: сведения о
библиотеках и их поступлении
в Библиотеку Академии наук 9 (1) 25

Шелов С. Д., Цумарев А. Э. О Третьем
Международном симпозиуме «Терми-
нология и знание» 9 (1) 33

Освалт Эмилия Б. Информационный и
терминологический поиск: практиче-
ский курс языка для профессиональной
аудитории [Рец на кн.] 10(1) 31

Черепуха Л.А. От библиографоведения
к документологии [Рец. на кн.] 11(1) 35

Авторский указатель

Акопян Д.А.	5(1)1	Архангельский Т. А.	4(2)24	Брежнева В.В.	3(1)12
Алешин Л.И.	3(1)16	Архипов М.В.	12(2)16	Буйлова Н.М.	2(1)30
	4(1)26				11(1)30
Алешин И.М.	4(1)15	Бабин М. А.	8(2)20	Булдакова Т.И.	6(1)16
Али Исмоили	4(1)20	Баяндин Н.И.	12(1)25	Бургучев С.С.	12(2)28
Алошги Х. Р.	4(2)1	Белкин С.В.	3(1)1		
Алпатова М.Г.	12(1)18	Белоногов Г.Г.	11(2)31	Васильченко К.А.	10(2)31
Андриянец-Буйко А.А.	2(2)15	Бениаминов Е. М.	3(2)1	Виноградов Д. В.	9(2)20
Антипов М.В.	12(1)18	Берёзкина Н.Ю.	8(1)33	Волкова А. Ю.	5(2)10
Аншаков О. М.	9(2)1	Бобров Л. К.	7(2)1	Волкова Т. А.	9(2)34
Арефьев П.Г.	8(1)1	Богословский М.М.	4(1)1	Волченков Е.Я.	4(1)9
Арутюнов В.В.	3(1)22	Боровик М.А.	8(1)16		
	7(1)10	Борщев В.Б.	11(2)5	Гайдамакин Н. А.	8(2)14
	9(1)6	Борюшкина Е. Н.	8(2)29	Гиляревский Р.С.	11(2)31

Голицына О. Л.	5(2)1	Максимов Н. В.	5(2)1	Смагин С. В.	1(2)9
Головин В.Ф.	12(2)16	Маркусова В.А.	6(1)30	Солошенко Н.С.	1(1)24
Гриняк В. М.	5(2)22	Мартirosян З. Г.	9(1)13	Сон Э. Е.	8(2)1
Громова Р.А.	1(1)13	Марушкина А. С.	4(2)13	Ставинский Е.Н.	1(1)13
	10(1)21	Мельникова Е.В.	2(1)19	Стародубов В. И.	8(1)1
Грузова А.А.	10(1)13		7(1)16	Строгонов В. И.	5(2)1
Гусев В. Д.	1(2)1	Михеенкова М.А.	10(2)1	Сугак Д.Б.	10(1)18
		Мокрозуб В. Г.	3(2)11	Сухорослов О.С.	4(1)15
Двоеносова Г. А.	9(1)1	Московкин В.М.	5(1)24	Сюнтюренок О.В.	6(1)1
	12(1)1		7(1)27		
Джалолов А. Ш.	6(1)16		8(2)9	Тихонов М.Н.	4(1)1
		Мошкин С. В.	2(2)15	Трофимов М. В.	5(2)22
Егоров В.С.	11(1)18	Мулева Ю. Н.	6(1)23	Тур О.Н.	10(2)21
Еляков А.Д.	5(1)1	Мыльников Л.А.	5(1)11	Тютюнник В. М.	7(2)18
Еркимбаев А. О.	8(2)1				
Ефременкова В.М.	1(1)24	Незнанов А. А.	5(2)16	Урсул А.Д.	2(1)1
		Немтинов К. В.	3(2)11		
Жиженко А. Б.	8(2)1	Нестеров А.В.	8(1)27	Фабрикантова Е.Ф.	12(2)1
Журавлев В.В.	12(2)16	Нешиной В. В.	5(2)27	Федосеева Л.Н.	10(2)27
				Финн В.К.	11(2)20
Забейайло М.И.	12(2)8	Окропишина О. В.	5(2)1		
Захарчук Т.В.	11(1)24	Опарина О.Д.	3(1)8	Хайруллин В. И.	8(2)33
Зицерман В. Ю.	8(2)1	Орлов И.А.	12(2)16	Халабия М.Л.	2(2)10
		Освалт Эмилия Б.	10(1)31	Ханжин А. Г.	9(1)19
Каменская М.А.	11(1)4	Осипов А.И.	2(1)30	Холодков К.И.	4(1)15
Карпунин И. Н.	5(2)16		11(1)30	Хорошилов А.А.	11(2)31
Касумов В. А.	2(2)1	Осипов А. Л.	7(2)1	Хорьков Д.А.	6(1)9
Касумова К. П.	2(2)1			Храмова Г.И.	1(1)13
Кириллова О.В.	1(1)24	Павловский В.Е.	12(2)16	Хренова Г.С.	8(1)33
Клещев А. С.	1(2)9	Пармон В. Н.	1(2)1		
Климова С.Г.	10(2)1	Пасичник Л.П.	3(1)26	Цветкова В.А.	2(1)19
Клышинский Э. С.	1(2)22	Петрина А. М.	7(2)7	Цветкова Л.А.	8(1)1
Кобзев Г. А.	8(2)1	Петров Т.Г.	2(2)15	Цумарев А. Э.	9(1)33
Ковтун В. А.	9(2)28		3(2)21		
Кожокару А. А.	9(1)19	Плешкевич Е.А.	1(1)30	Черепуха Л.А.	11(1)35
Кораблин Ю. П.	5(2)16		5(1)31		
Корягин В.Н.	4(1)15		7(1)1		
Кретов В.С.	12(1)25	Плющ М.А.	9(1)25	Шаронин К. А.	3(2)11
Кузнецов И.О.	12(2)36	Попов Д.А.	5(1)24	Шелов С. Д.	9(1)33
Кузьмин А. О.	1(2)1	Рахимова Н.М.	10(1)26	Шемакин Ю.И.	12(1)11
Кузьмина Д.А.	2(1)12	Романова М.С.	1(1)13	Шемберко Л.В.	8(1)16
Кураков Ф.А.	8(1)1		10(1)21		10(1)1
Куракова Н.Г.	8(1)1	Рудницкая Е. Л.	6(2)17	Шестерникова О. П.	5(2)10
			1(2)1	Шигорина Н.А.	5(1)24
Лапшин В. А.	2(2)24	Саломатина Н. В.		Шогин А.Н.	4(1)15
	3(2)1	Саркисян Д. Б.	9(1)13		12(1)18
	6(2)1	Семенюк Э.П.	1(1)1		
Лейчик В.М.	3(1)32	Сергеев А. Ю.	7(2)18	Ялпаев А. А.	8(2)14
Леонтьев С. В.	8(2)14	Сикорская О.Н.	8(1)33	Яцко В. А.	4(2)19
Леонтьева Т.М.	12(1)18	Силков С.В.	4(1)29		6(2)10
Лобанов А.С.	1(1)17	Ситникова И.С.	1(1)13		10(2)14
Логачева В. К.	1(2)22		10(1)21		
Лопатина Н.В.	3(1)19	Сладкова О.Б.	3(1)19		
	6(1)6	Слива А.И.	10(1)1		

РОССИЙСКАЯ АКАДЕМИЯ НАУК
Федеральное государственное бюджетное учреждение науки
ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
РОССИЙСКОЙ АКАДЕМИИ НАУК

предлагает научным работникам, аспирантам и другим специалистам в области естественных, точных и технических наук, желающим быстро и эффективно опубликовать результаты своей научной и научно-производственной деятельности, использовать способ публикации своих работ через *систему депонирования*.

«Депонирование (передача на хранение) – особый метод публикации научных работ (отдельных статей, обзоров, монографий, сборников научных трудов, материалов научных мероприятий – конференций, симпозиумов, съездов, семинаров) узкоспециального профиля, разрешенных в установленном порядке к открытому опубликованию, которые нецелесообразно издавать полиграфическим способом печати, а также работ широкого профиля, срочная информация о которых необходима для утверждения их приоритета. Депонирование предусматривает прием, учет, регистрацию, хранение научных работ и обязательное размещение информации о них в специальных информационных изданиях».

Подготовка и передача на депонирование научных работ происходит в соответствии с «Инструкцией о порядке депонирования научных работ по естественным, техническим, социальным и гуманитарным наукам» (М., 2003).

Результатом депонирования является публикация информации о депонированных научных работах в информационных изданиях ВИНТИ РАН – Реферативном журнале и аннотированном библиографическом указателе «Депонированные научные работы».

В соответствии с “Положением о порядке присуждения ученых степеней”, утвержденным Постановлением Правительства Российской Федерации от 30.01.2002 № 74 (в ред. Постановлений Правительства РФ от 20.04.2006 № 227, от 02.06.2008 № 424, от 20.06.2011 № 475), научные работы, депонированные в организациях государственной системы научно-технической информации, признаны публикациями, учитываемыми при защите кандидатских и докторских диссертаций.

Подать научную работу на депонирование можно обратившись в Отдел депонирования ВИНТИ РАН по адресу:

125190, Москва, ул. Усиевича, 20.
ВИНТИ РАН, Отдел депонирования научных работ.
Тел.: 8 (499) 155-43-28, Факс: 8 (499) 943-00-60.
e-mail: dep@viniti.ru

С инструкцией о порядке депонирования можно ознакомиться на сайте ВИНТИ РАН:
<http://www.viniti.ru>

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

ЦЕНТР НАУЧНО-ИНФОРМАЦИОННОГО ОБСЛУЖИВАНИЯ ВИНИТИ РАН

ПРЕДОСТАВЛЯЕТ КОПИИ ПЕРВОИСТОЧНИКОВ

ВИНИТИ РАН осуществляет обслуживание копиями первоисточников, хранящихся в фонде научно-технической литературы ВИНИТИ, в фондах других библиотек, а также в доступных ВИНИТИ электронных ресурсах.

Фонд научно-технической литературы ВИНИТИ включает более 2 млн изданий по точным, естественным и техническим наукам, в т.ч.:

- отечественные и иностранные периодические и продолжающиеся издания – с 1987 г.;
- отечественные книги – с 1987 г.;
- иностранные книги – с 1991 г.;
- рукописи, депонированные в ВИНИТИ, – с 1962 г.

Заказы на бумажные или электронные копии первоисточников принимает Центр научно-информационного обслуживания (ЦНИО) ВИНИТИ. ЦНИО ВИНИТИ обслуживает коллективных (организации и учреждения) и индивидуальных пользователей.

Формы обслуживания:

- абонементная (на основе договоров и предоплаты);
- разовые заказы (с предоплатой заказа по счету);
- индивидуальная форма обслуживания в читальном зале ЦНИО ВИНИТИ.

На сайте ВИНИТИ (<http://www.viniti.ru>) представлен полный Электронный каталог научно-технической литературы (<http://catalog.viniti.ru>), зарегистрированной в ВИНИТИ с 1994 г. Доступ для просмотра и поиска по Каталогу свободный. Постоянные абоненты ЦНИО ВИНИТИ, имеющие логин и пароль для работы с Каталогом, могут делать заказ копий непосредственно через Каталог.

Услуги по изготовлению копий первоисточников из фондов других библиотек предоставляются только постоянным абонентам. Место хранения первоисточников указывается в Электронном каталоге.

За подробной информацией обращаться по адресу:

125190, Россия, Москва, ул. Усиевича, 20, ВИНИТИ РАН. ЦНИО

Телефоны: 8 (499) 155-42-43, 155-42-09, 152-54-59

Факс: 8 (499) 943-00-60

E-mail: cnio@viniti.ru; **URL:** <http://www.viniti.ru>

БАЗА ДАННЫХ ВИНИТИ РАН

ВИНИТИ предлагает к использованию через WWW-сервер (<http://www.viniti.ru>) крупнейшую Федеральную базу отечественных и зарубежных публикаций по естественным, точным и техническим наукам. БД ВИНТИ РАН генерируется с 1981 г., обновляется ежемесячно, пополнение составляет около 1 млн документов в год. БД ВИНТИ представлена ретроспективными тематическими фрагментами и единой политематической БД (ретроспектива с 2001 г.), объединяющей все тематические фрагменты БД ВИНТИ.

БД ВИНТИ РАН в сети INTERNET

Сервер ВИНТИ – <http://www.viniti.ru> – обеспечивает on-line доступ к Базе данных ВИНТИ РАН круглосуточно без выходных.

На основе БД ВИНТИ РАН предоставляются следующие услуги:

- Диалоговый поиск научно-технической информации в **режиме on-line**;
- **Демо-версия**, позволяющая ознакомиться с основными функциями поисковой системы, составом данных, формами представления документов и получить навыки работы с системой;
- **Поисковые эксперты ВИНТИ** выполняют тематический поиск по разовым или постоянным запросам, а также окажут **консультационные услуги**.

БД ВИНТИ РАН на CD-ROM

Любые наборы тематических фрагментов БД ВИНТИ или их разделов могут быть предоставлены на **CD-ROM в поисковой системе (ИПС) "Сокол"**, обеспечивающей все поисковые функции, доступные в режиме on-line:

- Поиск можно вести в годовом или ретроспективном массиве (за несколько лет сразу) в одном или нескольких тематических фрагментах .
- Поиск по словам и любым словосочетаниям из заглавия, реферата, ключевых слов.
- Использование года, языка, рубрик, шифров тематических разделов БД для уточнения поиска.
- Поиск по словарю, выполняющему функции многоаспектного указателя, в том числе авторского, предметного, источников, индексов МПК, номеров патентных документов и депонированных рукописей и т.д.
- Возможность запоминания запросов для последующего использования и/или редактирования их.
- Чтение документов не только как в РЖ (последовательный просмотр документов одного номера за другим), но и чтение документов нужных тематических фрагментов (разделов) по оглавлению за весь период заказанной ретроспективы.

ИПС "Сокол" является прикладной программой Microsoft Windows.

Любые наборы тематических фрагментов БД ВИНТИ или их разделов могут быть подготовлены в **коммуникативных форматах** ISO-2709, МЕКОФ, txt на любых видах электронных носителей.

Продукты предоставляются на договорной основе.

Информационная служба БД ВИНТИ: 125190, Москва, ул. Усиевича 20, ВИНТИ

Телефон: (499) 155-45-01, 155-45-02, **Факс:** (499) 152-62-31 **e-mail:** csbd@viniti.ru