

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПО АТОМНОЙ ЭНЕРГИИ
МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ, НАУКИ И ТЕХНОЛОГИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ
РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ
МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНАЯ СЕССИЯ МИФИ-2004

НЕЙРОИНФОРМАТИКА–2004

**VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ Часть 2

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2004

УДК 004.032.26(06)

ББК 32.818я5

М82

НАУЧНАЯ СЕССИЯ МИФИ–2004. VI ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2004»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. Часть 2. – М.: МИФИ, 2004. – 200 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 28–30 января 2004 года в МИФИ в рамках VI Всероссийской научно-технической конференции «Нейроинформатика–2004».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор

Ю. В. Тюменцев, кандидат технических наук

ISBN 5–7262–0526–X © *Московский инженерно-физический институт
(государственный университет), 2004*

Содержание

Предисловие	6
<i>А. А. Жданов. О методе автономного адаптивного управления</i>	15
Введение	16
Имитационный метод автономного адаптивного управления	19
Проблемы создания систем ААУ и пути их реализации	26
Формирование и распознавание образов	26
Построение базы знаний и аппарата принятия решений . . .	30
Построение аппарата эмоций	34
Общая схема системы ААУ	40
Сравнительный анализ метода автономного адаптивного управле-	
ния и альтернативных подходов	42
Метод ААУ и нейросети	43
Метод ААУ и экспертные системы	46
Метод ААУ и системы нечеткой логики	48
Метод подкрепляющего обучения	49
Некоторые завершающие замечания	49
Заключение	53
Литература	53
<i>Л. А. Станкевич. Нейрологические средства систем управления ин-</i>	
<i>теллектуальных роботов</i>	57
Введение	58
Нейрологические средства систем управления	59
Грануляция	60
Ассоциативно-логическая обработка информации	61
Реализация базовых нейрологических модулей	69
Проекционная модель	70
Сеточная модель	72
Кластерная модель	74

Когнитивные нейрологические системы управления и агенты . . .	77
Интеллектуальные роботы и системы управления	88
Нейрологические системы управления для интеллектуальных ро- ботов	97
Система управления антропоморфного робота	97
Система управления гуманоидного робота	103
Заключение	107
Литература	109
С. А. Терехов. Нейро-динамическое программирование автономных агентов	111
Введение: Примеры систем с динамически-оптимальным управ- лением	112
Классическая задача оптимизации с ограничениями	114
Управляемая оптимальная динамика	115
Программное управление	115
Прогностическое управление на основе нейросетевой моде- ли системы	117
Нейро-динамическое программирование	120
Марковский процесс принятия решения	122
Уравнение Беллмана	125
Игровая стратегия и оценка позиции	127
Итерации в уравнениях Беллмана	128
Нейросетевая аппроксимация функции ценности	130
Алгоритм SARSA и Q -обучение	133
Примеры прикладных разработок	134
Динамическое управление портфелем финансовых активов .	135
Динамическое резервирование каналов сети сотовой связи .	136
Итоги	138
Литература	138
Н. Г. Макаренко. Как получить временные ряды из геометрии и то- пологии пространственных паттернов?	140
Введение	141
Вычислительная геометрия	144
Диаграмма Вороного	146
Триангуляция Делоне	147
Евклидово минимальное остовное дерево (ЕМОД)	149

Альфа-формы	151
Нейронный газ и топология близости	153
Геометрические вероятности: Игла Бюффона и парадокс Бертрана	158
Интегральная геометрия: Случайные точки и случайные прямые в R^2	162
Математическая морфология: Функционалы Минковского	170
Характеристика Эйлера	172
Характеристика Эйлера и персептроны	175
Стохастическая геометрия: Контурная статистика	177
Приложение методов контурной статистики	182
Вычислительная топология: Гомологии и числа Бетти	187
Числа Бетти для $H - \alpha$ карт	191
Эпилог	193
Литература	194

ПРЕДИСЛОВИЕ

1. В этой книге (она выходит в двух частях) содержатся тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 28–30 января 2004 года в МИФИ в рамках VI Всероссийской научно-технической конференции «Нейроинформатика–2004».

При отборе и подготовке материалов для лекций авторы и редактор следовали принципам и подходам, сложившимся при проведении трех предыдущих Школ (см. [1–5]). А именно, основной целью Школы было рассказать слушателям о современном состоянии и перспективах развития важнейших направлений в теории и практике нейроинформатики, о ее применениях. При этом особенно приветствовались лекции *междисциплинарные*, лежащие по охватываемой тематике «на стыке наук», рассказывающие о проблемах не только собственно нейроинформатики (т. е. о проблемах, связанных с нейронными сетями, как естественными, так и искусственными), но и о взаимосвязях нейроинформатики с другими областями мягких вычислений (нечеткие системы, генетические и другие эволюционные алгоритмы и т. п.), с системами, основанными на знаниях, с традиционными разделами математики, биологии, психологии, инженерной теории и практики.

Основной задачей лекторов, приглашаемых из числа ведущих специалистов в области нейроинформатики и ее приложений, смежных областей науки, было дать живую картину современного состояния исследований и разработок, обрисовать перспективы развития нейроинформатики в ее взаимодействии с другими областями науки.

Помимо междисциплинарности, приветствовалась также и *дискуссионность* излагаемого материала. Как следствие, не со всеми положениями, выдвигаемыми авторами, можно безоговорочно согласиться, но это только повышает ценность лекций — они стимулируют возникновение дискуссии, выявление пределов применимости рассматриваемых подходов, поиск альтернативных ответов на поставленные вопросы, альтернативных решений сформулированных задач.

2. В программу Школы-семинара «Современные проблемы нейроинформатики» на конференции «Нейроинформатика–2003» вошли следующие восемь лекций¹:

1. Я. Б. Казанович, В. В. Шматченко. Осцилляторные нейросетевые модели сегментации изображений и зрительного внимания.

¹ Первые четыре из перечисленных лекций публикуются в части 1, а оставшиеся четыре — в части 2 сборника «Лекции по нейроинформатике».

2. *В. Г. Яхно*. Нейроноподобные модели описания динамических процессов преобразования информации.
3. *А. Ю. Дорогов*. Быстрые нейронные сети: Проектирование, настройка, приложения.
4. *Н. Г. Ярушкина*. Нечеткие нейронные сети с генетической настройкой.
5. *А. А. Жданов*. О методе автономного адаптивного управления.
6. *Л. А. Станкевич*. Нейробиологические средства систем управления интеллектуальных роботов.
7. *С. А. Терехов*. Нейро-динамическое программирование автономных агентов.
8. *Н. Г. Макаренко*. Как получить временные ряды из геометрии и топологии пространственных паттернов?

Основные темы, рассматриваемые в этих лекциях — нетрадиционные² нейросетевые модели и их возможные применения.

3. Лекция **Я. Б. Казановича** и **В. В. Шматченко** «Осцилляторные нейросетевые модели сегментации изображений и зрительного внимания» посвящена теме, рассмотрение которой было начато в лекции *игумена Феофана (Крюкова)* на Школе 2002 года [6]. Эта тема относится к направлению в теории нейронных сетей, активно развиваемому в настоящее время. Оно ориентируется на изучение динамических и осцилляторных аспектов функционирования мозга. В рамках данного направления предложен ряд гипотез и моделей нейронных сетей, позволяющих объяснить возникновение пространственно-временных паттернов нейронной активности, а также их значение для обработки информации. Один из основных элементов этих моделей — *принцип синхронизации*, введение которого позволяет надеяться на решение ряда задач биологии и психологии, трудных для традиционных подходов. Лекция **Я. Б. Казановича** и **В. В. Шматченко** посвящена анализу моделей, в которых принцип синхронизации используется для решения задач сегментации объектов на изображении и формирования фокуса внимания.

4. К первой лекции по своему подходу и прикладной ориентации в определенной степени примыкает лекция **В. Г. Яхно** «Нейроноподобные модели описания динамических процессов преобразования информации», которая

²Нетрадиционные в том смысле, что отличаются от мультиперсептронов, сетей Хопфилда и еще двух-трех общеизвестных и повсеместно используемых видов моделей.

продолжает развитие темы, начатой автором в лекции на Школе 2001 года [7]. Здесь под *нейроподобными* понимаются распределенные системы, состоящие из активных элементов с несколькими устойчивыми или квазиустойчивыми состояниями, при этом взаимодействие между указанными неравновесными элементами осуществляется за счет нелокальных пространственных связей. Модели подобного рода успешно применяются для описания процессов в однородных нейронных сетях сетчатки глаза живых объектов, коры некоторых отделов головного мозга и т. п. Одна из важных областей применения предлагаемых моделей — адаптивные распознающие системы, а также совокупности таких систем, которые открывают возможность анализировать на количественном уровне реакции животных при восприятии и осознания действующих на них информационных сигналов.

5. Лекция А. Ю. Дорогова «Быстрые нейронные сети: Проектирование, настройка, приложения» базируется на том параллелизме, который существует между алгоритмом *быстрого преобразования Фурье* (БПФ), играющем огромную роль в обработке сигналов и многослойными нейронными сетями. Этот параллелизм заключается в том, что алгоритмы БПФ имеют выраженную многослойную структуру, которая подобна структуре многослойных персептронов. По этой причине представляется вполне естественным использовать потенциал, накопленный в области БПФ, для построения нейронных сетей соответствующей архитектуры, которые предлагается именовать быстрыми нейронными сетями (БНС). В лекции подробно рассматривается алгоритмическая сторона формирования БНС, которая иллюстрируется на многочисленных примерах, в первую очередь из области адаптивной фильтрации. Одна из интересных возможностей использования БНС обусловлена тем фактом, что структура БНС идентична структуре тензорных произведений векторных пространств. Из этого вытекает возможность использовать БНС для построения алгоритмов *квантовых вычислений*³. В лекции показано также, что парадигма БНС допускает многомерное обобщение, что открывает возможность создания быстродействующих классификаторов зрительных сцен.

³ Тематика квантовых вычислений и квантовых нейронных сетей уже затрагивалась в лекциях Школы (см., в частности, лекцию А. А. Ежова на Школе 2003 года [8]), а также в материалах Рабочего совещания «Квантовые нейронные сети» на конференции «Нейроинформатика-2000» [8].

6. Лекция **Н. Г. Ярушкиной** «Нечеткие нейронные сети с генетической настройкой» посвящена рассмотрению еще одной плодотворной парадигмы, которая позволяет сочетать методы обучения, характерные для нейронных сетей, с вербализацией правил вывода, типичной для нечетких систем. *Нечеткие нейронные сети* (ННС), изучаемые в рамках данной парадигмы, представляют собой реализацию систем нечеткого логического вывода методами нейронных сетей. Подобного рода сети включают в себя слои, которые состоят из специальных *И* и *ИЛИ* нейронов. Настройку нечетких нейронных сетей предлагается выполнять с помощью методов генетической оптимизации. В числе приложений для ННС, упоминаемых в лекции, задачи управления объектами (динамическими системами). Подобного рода комбинированное использование нескольких элементов, традиционно объединяемых под общим наименованием «мягкие вычисления» (искусственные нейронные сети, нечеткие системы, эволюционные вычисления) ранее в рамках Школы рассматривалось, в частности в лекциях **Ю. И. Нечаева** [10, 11] применительно к различным аспектам задачи управления динамическими объектами.

7. В той или иной степени с решением задач управления связаны следующие три лекции (**А. А. Жданова**, **Л. А. Станкевича** и **С. А. Терехова**). Общей чертой подходов, предлагаемых в этих лекциях, является то, что все они нацелены на решение задачи управления поведением объекта в условиях значительной неопределенности. В частности, в лекции **А. А. Жданова** «О методе автономного адаптивного управления» предлагается концептуальная модель нервных систем, названная методом «*автономного адаптивного управления*» (ААУ). Считается, что рассматриваемая система представляет собой совокупность из объекта управления, управляющей системы (УС) (моделируемой нервной системы) и среды. Данная УС является интеллектуальной, поскольку обладает такими свойствами, как наличие аппарата эмоций, который мотивирует, определяет, направляет и оценивает поведение УС; внутренняя активность, направленная на расширение знаний, повышающих вероятность выживания; адаптивность и саморазвитие; индивидуальность. На этой концептуальной основе предлагаются конкретные решения, позволяющие строить практически действующие управляющие системы, описаны примеры нескольких практических приложений.

8. Еще одна работа, связанная с проблемами интеллектуального управления, это лекция **Л. А. Станкевича** «Нейрологические средства систем управления интеллектуальных роботов». В ней вводится класс средств,

именуемых автором *нейрологическими*, которые способны обучаться в реальном времени отображению сложных функций и процессов. База для построения таких средств — нейронные сети и логические системы на правилах. Показано, что на основе нейрологических средств можно строить когнитивные и актуаторные структуры, способные обучаться, формировать и реализовывать сложное рациональное поведение динамических объектов в среде. Одна из очевидных сфер применения предлагаемых средств — создание интеллектуальных роботов. В качестве примеров такого применения указываются система управления для антропоморфного робота, а также перспективный проект системы интеллектуального управления гуманоидного робота. Лекторы Школы не впервые обращаются к тематике управления роботами, в том числе и антропоморфными роботами. В частности, эти проблемы рассматривались в лекциях А. А. Фролова и Р. А. Прокопенко [12], а также А. И. Самарина [13] на Школе 2001 года.

9. Еще один из подходов к управлению сложными динамическими системами представлен в лекции **С. А. Терехова** «Нейро-динамическое программирование автономных агентов», которая, в определенной степени, продолжает тематику, рассматривавшуюся в лекциях С. А. Терехова на Школах 2002 года [14] и 2003 года [15]. Здесь речь идет о многошаговых процессах принятия решений и об оптимизации таких процессов. Среди многообразия подходов, существующих в настоящее время для решения задач подобного рода, можно выделить приближенные методы поиска адаптационной стратегии, которые основаны на аппроксимации функций оптимального поведения искусственными нейронными сетями. Этот подход, использующий кроме нейросетей также еще и методологию динамического программирования, принято именовать *нейро-динамическим программированием*. Важную роль в нейро-динамическом программировании играет обучение нейронной сети на основе локальных подкреплений (reinforcement learning), используемое вместо традиционного обучения с учителем. Показана связь задачи целевой адаптации автономного агента при обучении с подкреплением и задачи прогностического оптимального управления. Приведены примеры приложений методов нейро-динамического программирования.

10. Лекция **Н. Г. Макаренко** «Как получить временные ряды из геометрии и топологии пространственных паттернов? Математические аспекты анализа распределенных динамических систем» продолжает серию его выступлений междисциплинарного характера, которые состоялись на Школах

2002 года [16] и 2003 года [17]. Объектом рассмотрения являются *распределенные динамические системы*, которые не только демонстрируют сложное поведение во времени, но и имеют нетривиальную пространственную структуру. И если в лекции [17] проекциями динамики рассматриваемых систем в «Мир экспериментатора» являлись одномерные временные ряды, то теперь это — «мгновенные снимки», изображения или «сцены». Таким образом для распределенных динамических систем приходится иметь дело с двумя видами сложности: временной, которая отслеживается временными рядами каких-либо интегральных параметров, и пространственной, которая «кодируется» геометрией и топологией «сцен». Применительно к данной ситуации рассматриваются пространственные паттерны, а также некоторые современные математические инструменты для «арифметизации» и анализа таких объектов, основанные на методах таких разделов математики, как вычислительная геометрия, геометрические вероятности, интегральная геометрия, математическая морфология, стохастическая геометрия, вычислительная топологии. Показано, что аппарат искусственных нейронных сетей вполне естественным образом взаимодействует со многими из этих элементов.

* * *

Для того, чтобы продолжить изучение вопросов, затронутых в лекциях, можно порекомендовать такой уникальный источник научных и научно-технических публикаций, как цифровая библиотека ResearchIndex (ее называют также CiteSeer, см. позицию [19] в списке литературы в конце предисловия). Эта библиотека, созданная и развиваемая отделением фирмы NEC в США, на конец 2002 года содержала около миллиона публикаций, причем это число постоянно и быстро увеличивается за счет круглосуточной работы поисковой машины.

Каждый из хранимых источников (статьи, препринты, отчеты, диссертации и т. п.) доступен в полном объеме в нескольких форматах (PDF, PostScript и др.) и сопровождается очень подробным библиографическим описанием, включающим, помимо данных традиционного характера (авторы, заглавие, место публикации и/или хранения и др.), также и большое число ссылок-ассоциаций, позволяющих перейти из текущего библиографического описания к другим публикациям, «похожим» по теме на текущую просматриваемую работу. Это обстоятельство, в сочетании с весьма эффективным полнотекстовым поиском в базе документов по сформулированному пользователем поисковому запросу, делает библиоте-

ку ResearchIndex незаменимым средством подбора материалов по требуемой теме.

Помимо библиотеки ResearchIndex, можно рекомендовать также богатый электронный архив публикаций [20], а также портал научных вычислений [18].

Перечень проблем нейроинформатики и смежных с ней областей, требующих привлечения внимания специалистов из нейросетевого и родственных с ним сообществ, далеко не исчерпывается, конечно, вопросами, рассмотренными в предлагаемом сборнике, а также в сборниках [1–5].

В дальнейшем предполагается расширение данного списка за счет рассмотрения насущных проблем собственно нейроинформатики, проблем «пограничного» характера, особенно относящихся к взаимодействию нейросетевой парадигмы с другими парадигмами, развиваемыми в рамках концепции мягких вычислений, проблем использования методов и средств нейроинформатики для решения различных классов прикладных задач. Не будут забыты и взаимодействия нейроинформатики с такими важнейшими ее «соседями», как нейробиология, нелинейная динамика (синергетика — в первую очередь), численный анализ (вейвлет-анализ и др.) и т. п.

Замечания, пожелания и предложения по содержанию и форме лекций, перечню рассматриваемых тем и т. п. просьба направлять электронной почтой по адресу tium@mai.ru Тюменцеву Юрию Владимировичу.

Литература

1. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // III Всероссийская научно-техническая конференция «Нейроинформатика-2001», 23–26 января 2001 г. / Отв. ред. Ю. В. Тюменцев. – М.: Изд-во МИФИ, 2001. – 212 с.
2. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. Ю. В. Тюменцев. Часть 1. – М.: Изд-во МИФИ, 2002. – 164 с.
3. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. Ю. В. Тюменцев. Часть 2. – М.: Изд-во МИФИ, 2002. – 172 с.
4. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая

- конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. Ю. В. Тюменцев. Часть 1. – М.: Изд-во МИФИ, 2003. – 188 с.
5. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. Ю. В. Тюменцев. Часть 2. – М.: Изд-во МИФИ, 2003. – 180 с.
 6. *Игумен Феофан (Крюков)*. Модель внимания и памяти, основанная на принципе доминанты // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2002. – с. 66–113.
 7. *Яхно В. Г.* Процессы самоорганизации в распределенных нейроноподобных системах: Примеры возможных применений // В сб.: «Лекции по нейроинформатике». – М.: Изд-во МИФИ, 2001. – с. 103–141.
 8. *Ежов А. А.* Некоторые проблемы квантовой нейротехнологии // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2003. – с. 29–79.
 9. Квантовые нейронные сети: Материалы рабочего совещания «Современные проблемы нейроинформатики» // 2-я Всероссийская научно-техническая конференция «Нейроинформатика-2000», 19–21 января 2000 г.; III Всероссийская научно-техническая конференция «Нейроинформатика-2001», 23–26 января 2001 г. / Отв. ред. А. А. Ежов. – М.: Изд-во МИФИ, 2001. – 104 с.
 10. *Нечаев Ю. И.* Нейросетевые технологии в бортовых интеллектуальных системах реального времени // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2002. – с. 114–163.
 11. *Нечаев Ю. И.* Математическое моделирование в бортовых интеллектуальных системах реального времени // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2003. – с. 119–179.
 12. *Фролов А. А., Прокопенко Р. А.* Адаптивное нейросетевое управление антропоморфными роботами и манипуляторами // В сб.: «Лекции по нейроинформатике». – М.: Изд-во МИФИ, 2001. – с. 18–59.
 13. *Самарин А. И.* Нейросетевые модели в задачах управления поведением робота // В сб.: «Лекции по нейроинформатике». – М.: Изд-во МИФИ, 2001. – с. 60–102.
 14. *Терехов С. А.* Нейросетевые аппроксимации плотности распределения вероятности в задачах информационного моделирования // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2002. – с. 94–120.
 15. *Терехов С. А.* Введение в байесовы сети // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2003. – с. 149–187.
 16. *Макаренко Н. Г.* Фракталы, аттракторы, нейронные сети и все такое // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2002. – с. 121–169.

17. Макаренко Н. Г. Эмбедология и нейропрогноз // В сб.: *«Лекции по нейроинформатике»*. Часть 1. – М.: Изд-во МИФИ, 2003. – с. 86–148.
18. Портал научных вычислений (Matlab, Fortran, C++ и т. п.)
URL: <http://www.mathtools.net/>
19. NEC Research Institute CiteSeer (also known as ResearchIndex) — Scientific Literature Digital Library.
URL: <http://citeseer.nj.nec.com/>
20. The Archive arXiv.org e-Print archive — Physics, Mathematics, Nonlinear Sciences, Computer Science.
URL: <http://arxiv.org/>

Редактор материалов выпуска,
кандидат технических наук *Ю. В. Тюменцев*

E-mail: tium@mai.ru

А. А. ЖДАНОВ

Институт системного программирования РАН, Москва

alexander.zhdanov@ispras.ru

О МЕТОДЕ АВТОНОМНОГО АДАПТИВНОГО УПРАВЛЕНИЯ¹

Аннотация

Рассматриваются основные положения бионического метода построения управляющих систем, названного *методом автономного адаптивного управления* (ААУ). Из общих для всех нервных систем свойств: дискретности строения и принципа действия, высокой неопределенности начальных знаний и приспособленности аппаратно-программной компоненты, а также необходимости осуществления управления и обучения в одном процессе, выводится принцип действия и строение управляющей системы. Предлагаются конкретные решения, позволяющие строить практически действующие управляющие системы. Решения описывают способы построения формальных нейронов, подсистем формирования и распознавания образов, базы знаний, принятия решений и аппарата эмоций. Описаны примеры нескольких практических приложений. Проведено сравнение предлагаемого подхода с альтернативными направлениями систем искусственного интеллекта. Показано, что предлагаемый подход может составить новое направление, которое целесообразно назвать «системы автономного искусственного интеллекта».

¹При поддержке РФФИ, проект № 03-01-00323

A. ZHDANOV

Institute for System Programming, Russian Academy of Sciences, Moscow
alexander.zhdanov@ispras.ru

ABOUT A TECHNIQUE OF AUTONOMOUS ADAPTIVE CONTROL**Abstract**

A bionic-based approach to control system design is suggested named as *autonomous adaptive control* (AAC) technique. Some basic concepts of this approach are considered. A principle of operation and structure of control system are derived from general nervous system properties such as discrete structure and functioning, initial knowledge uncertainty as well as necessity to combine control and learning processes within the same time period. Specific decisions are proposed, which allow to make real-world control systems. These decisions include techniques to build formal neurons, pattern generation and recognition subsystems, knowledge base, decision making and emotion subsystems. Some application problems are described solved by means of AAC technique. The AAC approach is compared with alternative artificial intelligence research areas. It is demonstrated that the approach suggested could be regarded as some new research area and could be named as “autonomous artificial intelligence systems.”

Введение

Задача моделирования живого мозга всегда вызывала повышенный интерес. Однако осознание чрезвычайной сложности объекта моделирования и выполняемых им функций ставило и продолжает ставить исследователей в тупик. Достаточно сказать, что мозг человека состоит из 10^{11} нервных клеток, соединенных 10^{14} связями, что средний размер нейрона менее 0.1 мм, а длина его отростка-аксона может превышать один метр при толщине в несколько микрометров, что один нейрон может быть соединен с сотнями и тысячами других нейронов при отсутствии какой-либо очевидной регулярности, чтобы представить трудности, которые стоят на пути решения данной задачи. Функции, выполняемые мозгом, также поражают своей сложностью. Задачи, которые решаются нервной системой даже самых

простых организмов, например, задача управления движением тела, могут быть воспроизведены в искусственных системах только с использованием таких сложных инструментов, как дифференциальное и интегральное исчисление, методы оптимизации и т. д., а решение их в реальном времени требует применения компьютеров. И уже за пределами понимания лежит тот факт, что этот же мозг одновременно легко решает и такие задачи, как формирование и распознавание образов, обучение и адаптация, генерирование целей и планов их достижения, принятие решений, в том числе на основе прогнозирования альтернативных вариантов и моделирования ситуаций в коллективе себе подобных посредством языка, а также другие задачи, которые в настоящее время получили лишь частичные и не связанные друг с другом решения.

Можно констатировать следующее:

1. В полной мере принцип действия мозга в настоящее время остается непонятным и не воспроизводится технически.
2. Имеющиеся в настоящее время методы решения задач, соответствующих некоторым функциям мозга, не имеют, по всей видимости, ничего общего с тем, как эти задачи решаются мозгом.

Основания для *вывода 1* заключаются в том, что:

- сегодня еще нельзя построить устройство, состоящее из 10^{11} параллельно работающих искусственных нейронов, каждый из которых осуществляет довольно сложную обработку информации;
- в настоящее время не существует убедительной функциональной модели нейрона, а также способов построения нейросетей, решающих задачи, свойственные мозгу;
- логическая организация мозга не понятна, даже на идейном уровне отсутствует представление о существовании таких проявлений мозга, как сознание, эмоции;
- передаваемая посредством генного аппарата информация отражает опыт предыдущих поколений и обеспечивает начальную приспособленность аппаратной, программной и информационной компонент мозга и всего организма к условиям его обитания, без которой невозможно решение задач выживания, управления, адаптации, накопления знаний. Вызывает сомнение, что длительный эволюционный отбор на протяжении уникальной истории можно заменить единовременным актом оптимизации системы при ее синтезе. Развитая методика организации эволюционного отбора также отсутствует.

Основания для *вывода 2* состоят в следующем. Точные науки и составляющие их объекты являются абстрактными образами, возникшими в человеческом мозге в результате развития его способностей к прогнозированию и выработке умственных моделей. Эти модели только отражают некоторые свойства природы и являются не более, чем инструментом, помогающим человеку найти и использовать закономерности природы. В реальной природе не существовало таких объектов, как дифференциалы, интегралы и т. п. Тем самым это есть новые объекты в природе, способные дать альтернативное решение некоторым задачам, ранее доступным только природе. Например, цапля обучается балансировать, стоя на одной ноге. Ее мозг решает задачу обучения и управления посредством нервных процессов, происходящих в сети нейронов, с участием органов и мышечного аппарата. Создав математический аппарат, человек получил возможность смоделировать это явление на основе дифференциального исчисления и построить искусственное устройство, способное балансировать на одной опоре. Теперь в природе имеется уже два объекта, способных решать эту сложную задачу балансировки — мозг и искусственное устройство. Но принцип работы у них совершенно разный. В мозге цапли дифференциальные уравнения не решаются ни в процессе управления, ни на этапе обучения. Несмотря на успехи математического моделирования, возможности построения управляющих систем на этой основе ограничены. Если в примере с цаплей одновременно потребуются воспроизвести и такие ее способности, как умение летать, решать навигационные задачи в длительных перелетах, общаться в стае, воспитывать потомство и т. д., то заведомо можно сказать, что точные науки тут не помогут из-за резкого усложнения математической модели.

Однако в настоящее время в природе появился третий объект, способный балансировать на одной опоре. Это устройство, управляемое нейросетью. Нейросеть — это пока еще очень простая модель небольшого фрагмента мозга. Тем самым это принципиально новое явление в природе — *искусственная нервная система*. От управляющей системы, построенной на основе классических математических наук, нейросеть отличается тем, что она без переделок может быть обучена и другим задачам, например, управлению полетом, распознаванию речи и т. п.

Очевидно, что задача приближения свойств современных нейросетей к свойствам мозга требует дальнейшего осмысления принципов работы мозга, его структуры как цельной системы, функций отдельных его подсистем и элементов. Этим определяется предмет исследования настоящей работы.

Практическая актуальность создания таких систем связана с возрастающим спросом на системы управления для объектов с плохо формализуемыми свойствами, требующими адаптации непосредственно в процессе управления.

В этом направлении получено много теоретических и практических результатов. Успехи прагматического направления исследований по искусственному интеллекту отодвинули в последние два десятилетия на второй план исходную задачу изучения управления в живых организмах, о которой говорил *Н. Винер* [1]. Имитационное направление, которому уделяли внимание такие исследователи, как *У. Кеннон* [2], *П. К. Анохин* [3], *А. А. Ляпунов* [4], *М. Месарович* [5] и многие другие, всегда являлось источником радикальных идей, которые прагматическое направление доводило в эволюционном порядке до практически полезных реализаций. Обратное взаимодействие прагматики на имитацию также является плодотворным, поскольку привносит новый математический и технический инструментарий, позволяющий строить и исследовать математические модели систем.

Отметим также, что, на наш взгляд, в настоящее время после пятидесятилетнего развития в прагматическом направлении формальных моделей нейрона и нейросети, предложенных *У. МакКаллоком* и *У. Питтсом* в 1943 году [6] и *Ф. Розенблаттом* в 1953 году [7] (направление получило название «искусственные нейронные сети» (ИНС)) наступает необходимость перехода к более адекватным действительности моделям нейрона, нервной системы и мозга. С одной стороны, накопился груз претензий к используемым в ИНС сильно упрощенным моделям нейрона и нейросети, претендующим, в лучшем случае, на простую модель небольшого регулярного участка нервной системы. С другой стороны, складывается впечатление, что в научном сообществе уже наработан и достаточно развит новый идейный, математический и программно-аппаратный инструментарий, который в совокупности может придать проблеме имитации новый импульс.

Имитационный метод автономного адаптивного управления

Здесь коротко представим основные положения концептуальной модели нервных систем, названной методом «автономного адаптивного управления» (ААУ) [8–30].

Пусть под управляющей системой (УС) понимается моделируемая нервная система, погруженная в организм — объект управления (ОУ), под системой пусть понимается совокупность УС, ОУ и среды.

Примем следующие *четыре исходных условия*, характерных для нервных систем.

1. «Условие автономности», под которым будем понимать только то обстоятельство, что УС является подсистемой ОУ, т. е. УС находится на борту ОУ и осуществляет управление на основе знаний, добываемых самостоятельно, взаимодействуя со своим окружением посредством блока датчиков (БД) и исполняющего органа (ИО). Тем самым

$$УС \subset ОУ \subset Среда = Система,$$

$$УС \cup БД \cup ИО = ОУ.$$

2. «Условие дискретности», которое отражает дискретность структуры УС (конечное множество нейронов, связей, датчиков, исполнителей) и принципа ее функционирования (дискретность нервных импульсов, образов — как элементов информации, моментов времени). При этом возможен непрерывный характер изменения некоторых параметров, таких как размеры синапсов, частотные характеристики импульсных последовательностей.
3. «Условие максимальной начальной приспособленности» отражает наличие приспособленности ОУ и УС к усредненным условиям жизни ОУ в данной среде в результате действия механизмов типа естественного отбора, что определяет типы датчиков и исполнителей, классы потенциально возможных в данной системе образов, оценки качества важнейших для ОУ образов, и т. п. При синтезе ОУ и УС процесс естественного отбора, возможно, может быть заменен максимальным использованием априорной информации.
4. «Условие минимума исходных знаний» отражает наличие информационных пространств, которые должны быть заполнены знаниями, найденными УС в процессе функционирования в реальной системе. Это условие соответствует наличию неопределенности свойств системы, максимальной для УС в момент начала ее функционирования.

Еще один необходимый элемент — *целевые функции* УС, которыми должны быть:

- выживание ОУ;
- накопление знаний.

Эти две целевые функции взаимосвязаны в том отношении, что достижение одной из них повышает вероятность достижения другой.

Из сказанного вынужденно следует представление о системе (рис.1), в котором можно видеть ОУ погруженным в среду, УС погруженной в ОУ, а также все возможные маршруты распространения воздействий в системе (помеченные буквами стрелки на рисунке). Пусть каждый из этих макрообъектов оказывает воздействия на систему через свой выход, каждый из макрообъектов пусть воспринимает воздействия системы через свой вход. Кроме того, в системе важно наличие источников случайных воздействий (белые кружки на рисунке — «истoki») и мест поглощения воздействий (черные кружки на рисунке — «стоки»).

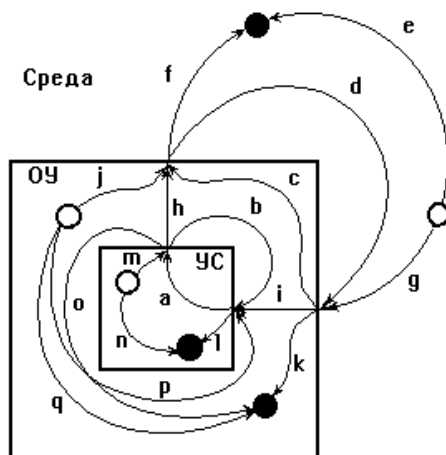


Рис. 1. Общая структура системы

Очевидно, что для достижения своих целевых функций УС должна найти те из воздействий h , которые образуют цикл «управляемого взаимодействия»

$$h \rightarrow d \rightarrow i \rightarrow a \rightarrow h \rightarrow \dots,$$

зафиксировать информационное отображение цикла УВ в своей памяти — базе знаний (БЗ), оценить полезность тех или иных элементов знаний и использовать эти знания для выживания, одновременно прилагая усилия для получения и накопления новых знаний. Здесь a это информационный процесс в УС, h — процесс преобразования информационных команд в физические воздействия, d это различные процессы в окружающей среде, i — процесс преобразования части входных воздействий в информационные входные сигналы для УС. Наличие в системе истоков и стоков вносит в управляемое взаимодействие многочисленные случайные компоненты и приводит к потерям информации. Накапливая знания, УС стремится к уменьшению неопределенности в своем отображении управляемого взаимодействия.

Подобное макроописание позволяет понять задачу, которую должна решать УС. В общем виде можно видеть следующую ее постановку. На заданное множество входных полюсов УС $v_1, v_2, \dots, v_i, \dots, v_n$ (например, это n бинарных выходов БД) поступает входной поток информации. Пусть, например, это будет эквидистантная во времени последовательность двоичных векторов $\mathbf{v}_a(t_1), \mathbf{v}_b(t_2), \dots, \mathbf{v}_c(t_k), \dots, \mathbf{v}_d(t)$, где t — текущий момент времени. Если последовательность не эквидистантная, то должны быть средства синхронизации потоков данных. Согласно рис. 1, семантически каждый вектор $\mathbf{v}_c(t_k)$ может содержать информацию, пришедшую ко входу УС по маршрутам i, b и/или p . Здесь i это информация, поступившая с датчиков, b — информация, отражающая действия, совершенные УС, p это незакономерные помехи от истоков в ОУ (информация от i и b также содержит случайные компоненты). Информация, поступающая по маршрутам i, b и p , может отображаться на определенные подмножества компонент вектора $\mathbf{v}_c(t_k)$.

Первая задача, которую должна и может решать УС, состоит в нахождении неслучайных регулярных пространственно-временных комбинаций компонент в потоке входных векторов $\mathbf{v}_a(t_1), \mathbf{v}_b(t_2), \dots, \mathbf{v}_c(t_k), \dots, \mathbf{v}_d(t)$. Произвольные примеры трех таких образов показаны на рис. 2.

Если УС обнаруживает наличие такой регулярности, то УС должна:

- запомнить информацию о ней как самостоятельный объект — образ,
- уметь распознавать прообраз этого образа во входной информации (регулярность, которая привела к формированию образа) при его последующих появлениях,
- при накоплении достаточной статистической надежности — уметь распознавать образ протяженного во времени прообраза раньше, чем

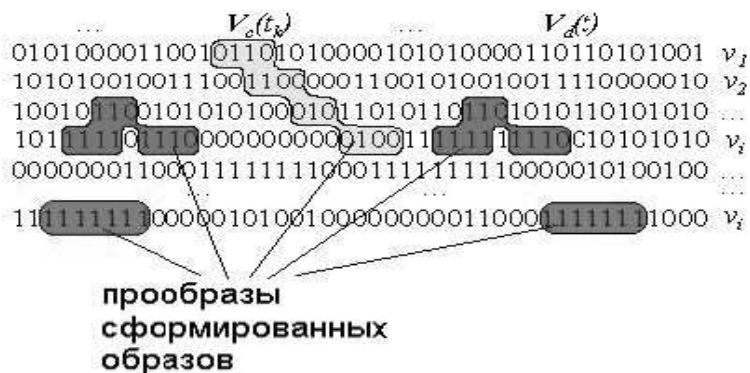


Рис. 2. Примеры неслучайных регулярных пространственно-временных комбинаций компонент в потоке входных векторов

закончится его наблюдение на входных полюсах, а также распознавать при наличии помех,

- оценить соответствие этого образа целевой функции выживания ОУ.

Путь к решению последней задачи мы видим в организации в УС специальной подсистемы, которая эквивалентна *аппарату эмоций* в живом организме. Это многофункциональная подсистема, основная задача которой — соотносить отвлеченные информационные сигналы, например, сигналы о распознавании образов, отображающих текущие состояния ОУ, с объективной полезностью или опасностью их прообразов для выживания ОУ. Безусловно, такая оценка для важнейших состояний ОУ (температура среды, количество поглощенной пищи и т. п.) может быть найдена опытным путем в эволюционной предыстории предков ОУ, а для искусственных ОУ — выведена из априорной информации. Оценки для образов, которые связаны с конкретными условиями существования данного ОУ, могут быть найдены УС опытным путем посредством корреляционного анализа фактов распознавания этих образов и значений некоторой текущей интегральной оценки состояния ОУ, которую обозначим S^t . В свою очередь, текущее значение оценки S^t зависит от оценок всех распознанных в текущий момент образов. Формальное описание способов получения таких оценок (фактически — аппарата эмоций) дано в [21]. Здесь обратим внимание на

две важных функции аппарата эмоций. Это закрепленное в УС стремление к получению все более высоких оценок S^t , что является причиной постоянной активности УС, направленной на достижение обеих указанных целевых функций. Кроме того, текущее значение оценки S^t определяет целесообразную глубину просмотра базы знаний при принятии решений в текущей ситуации (для быстрого принятия решения в угрожающей ситуации можно не анализировать второстепенные последствия выбираемого действия — образы с меньшими оценками).

Пусть УС сформировала некоторый образ, запомнила его в памяти сформированных образов в виде объекта, который обозначим O_k , и в некоторый момент t распознает его, что обозначим как $O_k^t = 1$ (если сформированный образ O_k не распознан в момент t , то полагаем $O_k^t = 0$). Пусть, например, прообразом данного образа является некоторый пространственно-временной процесс длительностью T тактов, но УС распознала образ уже через T' тактов после его начала t_{ks} , и $T' < T$. Если семантически прообраз данного образа в векторах $\mathbf{v}_a(t_{ks}), \mathbf{v}_b(t_{ks+1}), \dots, \mathbf{v}_c(t_{ks+T'}), \dots, \mathbf{v}_d(t_{ks+T})$, представлял собой отображение информации, пришедшей на вход УС по маршруту i (см. рис. 1), то это сравнительно простой случай, интересный в основном тем, что УС имеет прогноз, в частности, изменения оценки своего состояния S^t к моменту, когда завершится наблюдение всего прообраза. Одновременно может быть распознано некоторое множество образов. Однако ситуация становится более интересной, если в прообразе распознанного образа имел место хотя бы один вектор $\mathbf{v}(t)$, отражающий информацию, пришедшую к УС по маршруту b , т. е. в прообраз входило некоторое действие, совершенное УС (например, это сигнал от рецепторов о том, что была сокращена такая-то мышца в результате соответствующей команды). Обозначим такое действие символом Y_l . Тогда прообраз можно разбить на три последовательных во времени составляющих:

- i) информация о ситуации a ,
- ii) информация о действии l ,
- iii) информация о ситуации b .

Поскольку, по определению, прообраз сформированного образа — явление не случайное, то данную тройку можно интерпретировать как импликацию

$$\text{условие} \rightarrow \text{действие} \rightarrow \text{следствие}.$$

Очевидно, что в данной ситуации могло быть распознано два или более



Рис. 3. Пример ситуации, когда распознано два или более образов, имеющих одинаковое условие, разные действия и следствия

образов (рис. 3), имеющих одинаковое условие (ситуацию a), разные действия (действие m) и, возможно, разные следствия (ситуацию c).

Итак, если в некоторый момент

- УС одновременно распознает несколько временных образов, при этом
- распознавание наступило раньше окончания наблюдения прообразов,
- в составе прообразов ожидаются альтернативные действия, ведущие к альтернативным последствиям,
- распознанные образы имеют свои оценки,

то появляется возможность принятия решения на основании сравнения ожидаемых приращений интегральной оценки качества состояния S^t для альтернативных действий [10, 11].

Если будущий результат действия, предвидимый УС в текущий момент, влечет распознавание образов, которые могут быть распознаны по результатам планируемого действия, т. е. причинно-следственная цепочка образов и действий продолжается на несколько шагов в будущее, а УС при этом обладает языковыми средствами для манипулирования с этими образами и действиями из своей базы знаний, то появляются основания говорить об *интеллекте* [13].

Поскольку происходит формирование новых образов, то возможна ситуация, когда альтернативные действия, приносящие ранее одинаковый

конечный результат — распознавание одинаковых образов, начнут различаться в отношении новых сформированных образов, т. е. более тонких последствий выбираемого действия. Учет этих различий делает действия УС со временем все более точными, а их последствия — все более предсказуемыми. Это одна из возможностей *саморазвития* УС.

Рассмотренная интеллектуальная управляющая система обладает следующими *основными свойствами*.

- Поведение УС мотивируется, определяется, направляется и оценивается *аппаратом эмоций*, целевая функция которого содержит в себе явно или не явно цели выживания и накопления знаний.
- УС обладает *внутренней активностью*, направленной на расширение знаний, повышающих вероятность выживания.
- УС обладает свойствами *адаптивности* и *саморазвития*. Именно, формируются и оцениваются новые образы, формируются и уточняются новые действия, увеличивается глубина прогнозирования последствий альтернативных вариантов действий и др.
- Знания УС сильно зависят от ее индивидуального опыта, в том числе от случайных событий, т. е. УС обладает своей *индивидуальностью*.

Проблемы создания систем ААУ и пути их реализации

Как видно из сделанного качественного описания рассматриваемых систем, их реализация сопряжена с необходимостью решения задач, некоторые из которых здесь перечислим.

Формирование и распознавание образов

Задача *формирования и распознавания образов* (ФРО) в потоке входной многоканальной информации. В общем виде на вход ОУ поступают непрерывные воздействия различной физической природы (зрительные, звуковые и т. д.). Аппарат ФРО должен в процессе самообучения сформировать множество дискретных образов, соответствующих классам пространственно-временных регулярностей и далее распознавать такие образы. Распознаваемые образы порождают формирование образов над образами, т. е. образов более высокого порядка. Переход от континуальных величин к дискретным должен происходить уже в БД. В нетривиальных случаях при отсутствии априорного алфавита классов такого рода задача является трудной.

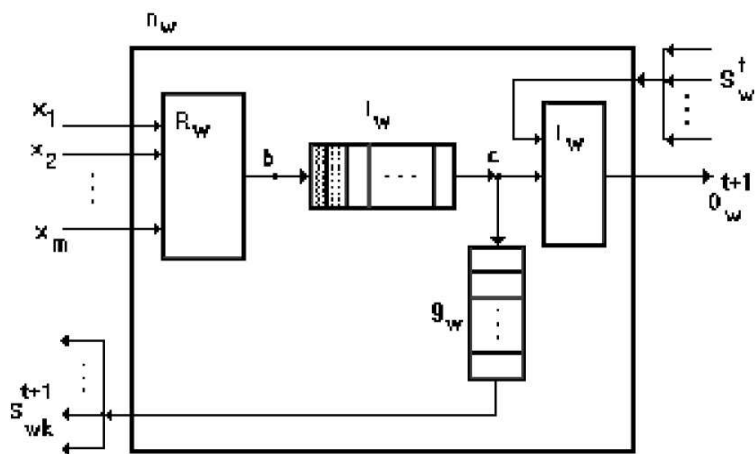


Рис. 4. Основная модель нейрона в методе ААУ

В некоторых условиях для построения БД и/или начальных трактов аппарата ФРО возможно применение персептронов и генетического подхода, либо использование аппарата эмоций в качестве учителя. Напомним, что задача распознавания образов даже таких простых регулярностей, как периодические компоненты во временных рядах, требует применения сложного математического аппарата анализа временных рядов. Мы полагаем, что поиск неслучайных составляющих во входном многоканальном потоке входных данных должен основываться на аппарате корреляционного анализа, тогда решение о формировании нового образа может приниматься при накоплении достаточных статистических оснований.

В естественных системах аппарат ФРО строится на основе нейронов. Очевидно, что нейрон должен и может накапливать статистику по наблюдаемым входным векторам, изменять свой способ функционирования при появлении достаточных статистических оснований, говорящих о неслучайной природе прообраза, т. е. формировать образ и быстро распознавать уже сформированный образ. Нами разработано несколько формальных нейронов с такими свойствами [12, 14, 15, 17, 18, 23, 25, 29]. Описание основной модели нейрона (рис. 4) состоит в следующем.

На вход нейрона n_w в момент t поступает двоичный вектор

$$\mathbf{X}^t = (x_1, x_2, \dots, x_i, \dots, x_m)$$

и сигнал S_w^t . В момент $t + 1$ нейрон производит выходные сигналы O_w^{t+1} и S_{wk}^{t+1} согласно логическим выражениям

$$O_w^{t+1} = \neg S_w^t \& [(b_w^t \& l_w^t) \vee O_w^t] \quad \text{и} \quad S_{wk}^{t+1} = b_w^t \& l_w^t \& g_w^t.$$

Значение сигнала b_w^t в точке « b » определяется зависимостью $b_w^t = 1$, если $h_w/m \geq p(N^t)$ и $b_w^t = 0$ в других случаях. Здесь h_w есть число таких компонент x_i вектора \mathbf{X}^t , которые имеют значение 1 в момент t ; N^t есть число событий $b_w^t = 1$ в предыстории этого нейрона от $t = 0$ до t ; $p(N)$ есть сигмоидальная функция. Последняя определена для значений $N = 0, 1, \dots$ и уменьшается от некоторого значения $p(0) = p_{max}$, $p \leq 100\%$ до значения $p(\infty) = p_{min}$, $p_{min} < p_{max}$. Кроме того, $p(M) = p_M$, $p_{min} < p_M < p_{max}$, где M есть константа. Значения p_{min} , p_M , p_{max} и M задаются для каждого нейрона априори. Переменная l_w^t показывает состояние элемента l_w в момент t и может принимать значения 0 или 1 согласно условию: $l_w^t = 0$, если $N^t < M$ и $l_w^t = 1$ — иначе.

Элемент T подобен триггеру, который переключается сигналом

$$(b_w^t \& l_w^t) = 1$$

(см. точку « c » на рис. 4) в состояние, при котором выходной сигнал O_w^{t+1} становится равным 1, и сигналом $S_w^t = 1$ переключается в состояние, при котором выходной сигнал O_w^{t+1} принимает значение 0. Переменная g_w^t определяется условием: $g_w^t = 0$, если $Z^t < L$ и $g_w^t = 1$ — в противном случае, где Z^t есть число единичных сигналов, наблюдавшихся в точке « c » в течение предыстории. Константы L определены для каждого нейрона.

Векторы \mathbf{X}^t , для которых $b_w^t = 1$, «обучают» нейрон (число N увеличивается). Число M для нейрона подобрано так, чтобы N не превысило M за время жизни нейрона, если такие векторы \mathbf{X}^t есть случайные явления. С другой стороны, число N достигнет величины M в случае, если этот вектор есть неслучайное явление в системе (с заданной вероятностью ложной тревоги). Если событие $N^t = M$ случится с нейроном n_w , то мы говорим, что нейрон n_w *обучен* и образ O_w сформирован, начиная с этого момента t . Необратимый процесс роста N от 0 до M в нейроне n_w есть *процесс обучения* нейрона n_w и, тем самым, *процесс формирования образа*

O_w . Если образ сформирован, то он не может уже быть «расформирован» (переучивание УС происходит за счет доучивания; память о прежних образах и знаниях сохраняется в обученных нейронах). Сформированный образ может быть распознан в текущий момент ($O_w^{t+1} = 1$) или может быть не распознан ($O_w^{t+1} = 0$). Несформированный образ не может быть распознан.

Сигнал $O_w^{t+1} = 1$ может быть отключен сигналом $S_w^t = 1$ только после того, как информация этого сигнала будет использована УС.

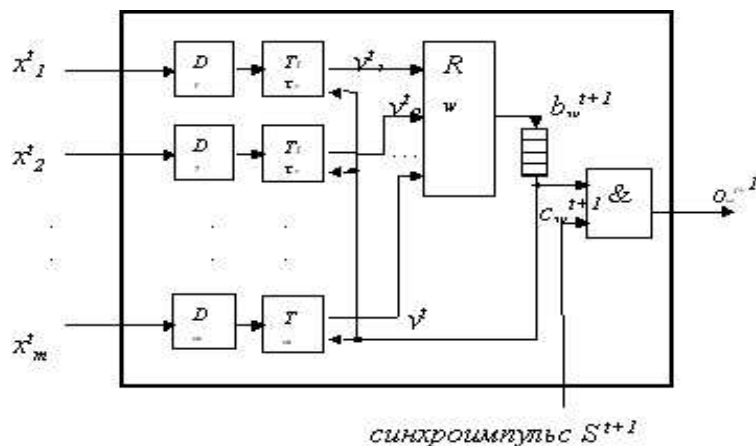


Рис. 5. Модификация основной модели нейрона, учитывающая синаптические и другие возможные задержки D_i на отдельных синапсах, а также актуальную длительность τ_i сигнала

Итак, эта модель нейрона содержит три необратимо обучаемых элемента, именно R_w , l_w и g_w , а также один триггерный элемент T_w . Выходной сигнал нейрона генерируется на следующем такте после получения входного вектора. Поскольку в общем случае нейрон может быть соединен не со всеми нейронами предыдущего слоя (см. ниже, на рис. 6), то прообразом образа является пространственно-временная последовательность сигналов на входе нейросети. Входы данного нейрона могут не иметь весов, в этом случае нейрон различает только единичные векторы с учетом полноты их состава, задаваемой функцией $p(N)$. Нейрон обучается под воздействием единичных (с учетом функции $p(N)$) входных векторов, другими словами,

нейрон способен обнаружить коррелирующие единичные сигналы, если они подаются на входы нейрона.

На рис. 5 показана модификация данного нейрона, учитывающая синаптические и другие возможные задержки Δ_i на отдельных синапсах, а также актуальную длительность τ_i сигнала, которая может быть связана, например, с шириной и скоростью прохождения волны деполяризации в месте аксонного холмика, и другими причинами, определяющими характерную длительность импульса, воспринимаемого элементом R_w как единичный сигнал. Будем полагать, что характерная длительность возбуждения, соответствующая единичному выходному сигналу $y_i = 1$ триггерного элемента T_i , поддерживается не дольше, чем в течение времени τ_i . Величины задержек Δ_i и τ_i задаются при синтезе УС. Обучаться будет тот нейрон, у которого величины задержек совпадают с характером входного сигнала.

Следующая модификация нейрона связана с моделированием известного факта ускорения роста тех синапсов, по которым приходят коррелирующие входные сигналы [17, 25]. Это позволяет нейрону реагировать не только на единичный вектор, но на любую неслучайную комбинацию единичных входных сигналов.

Итак, сеть, собранная из рассмотренных нейронов (небольшой фрагмент сети показан на рис. 6), способна решать задачу ФРО. При этом априорная информация определяет топологию сети, а неопределенность требует соответствующего избытка нейронов. В сети обучаются те нейроны, на которые отображаются пространственно-временные закономерности входного потока информации.

Построение базы знаний и аппарата принятия решений

Поскольку рассмотренный нейрон фактически обнаруживает коррелирующие сигналы, на его основе можно построить БЗ, где отдельный нейрон фиксирует рассмотренную выше тройку

⟨образ условия, образ действия, образ следствия⟩

(см. [9, 11, 12, 14–17, 20]). В элементарном случае БЗ может быть представлена трехмерной матрицей нейронов, измерения и размерности которой соответствуют образам указанной тройки множеств. Принятие решения осуществляется обученными нейронами, что становится возможным при уменьшении порога $p(N)$.

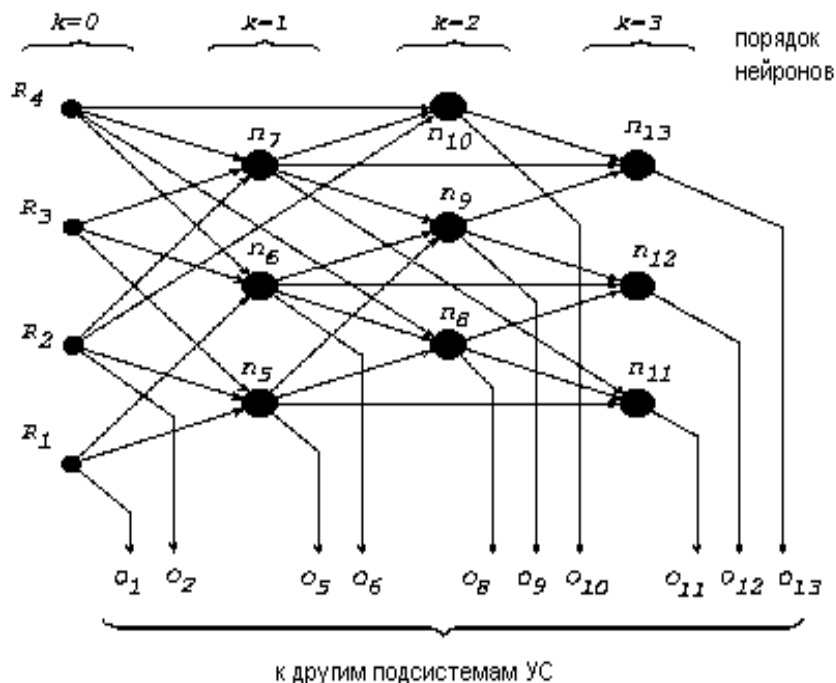


Рис. 6. Небольшой фрагмент подсети ФРО, в которой нейроны соединены нерегулярным способом, что обеспечивает формирование образов пространственно-временных явлений

Рассмотрим простейший способ построения такой БЗ с помощью нейронов, показанных на рис. 4. Как отмечалось выше, такой нейрон не только может обнаруживать и запоминать факт корреляции единичных входных сигналов, но и при снижении значений функции $p(N)$ распознавать сформированный образ при наблюдении неполного прообраза с заданной вероятностью ложной тревоги. Эти два свойства нейрона можно использовать для построения БЗ. Представим элементарное знание последовательностью событий вида

$$O_w^{t-1} \& Y_k^{t-1} \& O_v^t, \quad (1)$$

где O_w^{t-1} есть образ, распознанный в момент $t-1$ (это «образ исходного

состояния»), Y_k^{t-1} есть действие, совершенное в момент $t-1$, а O_v^t есть образ, распознанный в момент t (это «образ результата»). Такие последовательности записываются в БЗ в определенной структуре. Покажем, как такую последовательность можно обнаружить, запомнить и использовать посредством нейрона. Соединим входы нейроны так, как показано на рис. 7 (нейрон изображен одним прямоугольником).

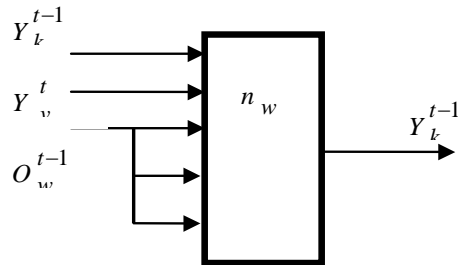


Рис. 7. Обнаружение, запоминание и использование последовательности O_w^{t-1} & Y_k^{t-1} & O_v^t с помощью нейрона

Если данный нейрон n_w получит нужное число раз на свои m входов единичные сигналы, т. е. наблюдает прообраз вида (1), все сигналы O_w^{t-1} , Y_k^{t-1} , O_v^t которого имеют значение 1, то он обучится. Если после обучения нейрона, он нужное число раз распознает образ (1), то у данного нейрона произойдет снижение пороговой функции $p(N)$. Когда порог $p(N)$ снизится до величины $m-1$, данный нейрон может срабатывать уже при подаче только двух единичных сигналов: $O_v^t = 1$ и $O_w^{t-1} = 1$. Эти свойства можно использовать для:

- обнаружения и фиксации закономерных в данной системе прообразов вида (1) на этапе обучения УС;
- для отыскания на этапе управления тех нейронов в подсети БЗ, которые обнаружили и зафиксировали существующие в данной системе прообразы вида (1).

Организуем БЗ в виде трехмерной сетки (куба) нейронов (рис. 8), оси которой будут соответствовать всем возможным образам и действиям O_w^{t-1} , Y_k^{t-1} , O_v^t . В каждом узле куба разместим нейрон так, как это показано на рис. 7.

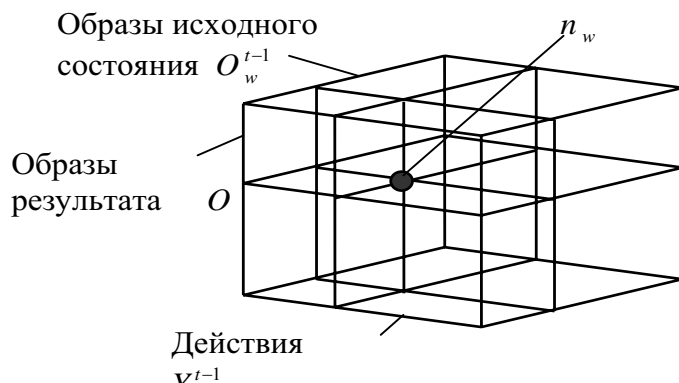


Рис. 8. Организация БЗ в виде трехмерной сетки (куба) нейронов

В процессе обучения в кубе БЗ обучаются нейроны, соответствующие свойствам конкретной системы. На этапе принятия решений в момент t смысл величин по осям куба БЗ сменяется с O_w^{t-1} , Y_k^{t-1} , O_v^t на O_w^t , Y_k^t , O_v^{t+1} , соответственно. В момент t распознаны «образы исходных состояний» по оси O_w^t . Блок выбора действий УС по очереди подает единичные сигналы на входы O_v^{t+1} «образов результата» в некотором порядке. Те нейроны, которые обучены, будут выдавать единичные сигналы на выходы Y_k^t . Необученные нейроны будут выдавать нулевые выходные сигналы. Таким способом будут обнаружены все возможные в текущей ситуации действия Y_k^t . Блок принятия решений теперь может выбрать те из возможных действий, которые лучше соответствуют заданным целевым функциям УС. Например, если «образы результата» O_v^{t+1} , упорядочить по их качественным оценкам, то блок принятия решений может перебирать эти образы в порядке уменьшения их качественной оценки. Первое найденное действие и следует принять к исполнению, так как именно оно, согласно БЗ, приведет к лучшему из возможных в текущей ситуации результатов.

Проблема, которая здесь может возникнуть в конкретном практическом приложении, связана с большим количеством нейронов, необходимых для заполнения такого куба БЗ. Возможны следующие подходы к уменьшению числа необходимых нейронов.

- Из априорной информации о заданном приложении находятся с заданной уверенностью те участки куба БЗ, которые не будут содержать обученных нейронов. Такие участки куба заранее отсекаются. Этот подход моделирует естественный отбор и соответствует условию максимальной исходной приспособленности.
- Для построения БЗ можно использовать нейроны, анализирующие состав входных векторов не только из класса единичных векторов, и запоминающие номера входов, по которым приходят коррелирующие сигналы (пример такого нейрона (нейрон типа III) описан в [18, 25]). Это даст возможность перейти от трехмерной к двумерной матрице БЗ, с соответствующим сокращением числа нейронов.
- Возможна разработка алгоритма перенастройки БЗ в процессе обучения УС. В этом случае в рамках заданных ресурсов, ограничивающих множество нейронов, сначала грубо определяются области БЗ, в которых происходит обучение. Затем на основе методов статистических решений принимается решение об отсечении некоторых областей и перераспределении множества нейронов в пользу наиболее важных областей БЗ, что даст возможность проводить более точное обучение. Затем процедура повторяется.
- Возможно применение генетических алгоритмов для формирования нейросети БЗ, что будет заменять собой поиск оптимальной структуры БЗ в процессе естественного эволюционного отбора на множестве поколений.

Построение аппарата эмоций

Мы предлагаем рассматривать аппарат эмоций организма как основной системообразующий фактор, который определяет строение и функционирование нервной системы и является одновременно: источником внутренней активности, целевой функцией управления, средством качественного оценивания общего текущего состояния и отдельных элементов знаний — образов и действий, внутренними часами системы, а также каналом передачи оценок качества информации при общении [21]. По нашему убеждению основные функции аппарата эмоций в живом организме состоят в следующем.

А. Аппарат эмоций является источником внутренней активности нервной системы. В живом организме должен иметь место некий источник «энергии» внутренней активности. Это должна быть универсальная «движущая сила», заставляющая активно действовать любой организм, независимо от его сложности, возраста и интеллекта. Здесь также следует различать две цели: цель «создателя» организма, и целевую функцию, «защитную» в конструкции организма. Мы видим в аппарате эмоций механизм, задающий активное поведение.

В аппарате эмоций, прежде всего, имеет место множество объектов, раздражение которых сопровождается приятными, либо неприятными для организма ощущениями, включая самые предельные их степени. Это множество ощущений и их носителей-объектов упорядочено по степени этого качества — «приятности» ощущения. Пусть в каждый момент времени выделен один и только один из этих объектов, который назовем *эмоциональной оценкой текущего состояния системы*, и обозначим как $S(t)$. Соответственно, эта оценка может принимать одно из следующего множества возможных значений

$$S = \{s_1, s_2, \dots, s_z, \dots, s_Z\},$$

где, например, $s_1 = -5$, $s_2 = -4$, ..., $s_Z = 5$. Значение $S(t) = s_z \in S$ функционально зависит от совокупности эмоциональных оценок $p_{z,j}$ множества распознанных в текущий момент образов O_j . Здесь возможные значения эмоциональных оценок образов также могут быть равны $p_{1,j} = -5$, $p_{p2,j} = -4$, ..., $p_{Z,j} = 5$. В множестве образов, которые система уже научилась распознавать, что составляет ее «память образов», распознанные в текущий момент образы пометим символом «1», т. е. если образ O_j распознан в момент t , то будем писать $O_j(t) = 1$, а если не распознан, то будем писать $O_j(t) = 0$. Эмоциональная оценка $p_{z,j}$ образа является атрибутом образа. Пусть функция, определяющая оценку $S(t)$, есть, например, просто средняя величина от эмоциональных оценок всех распознанных в текущий момент образов, округленная до ближайшего значения из множества S . Такой аппарат можно представить в виде своего рода термометра, показывающего среднюю текущую «эмоциональную температуру тела».

Жестко «зашьем» в управляющую систему целевую функцию максимизировать *эмоциональную оценку текущего состояния системы* $S(t)$. Другими словами, организм должен бояться и избегать неприятных эмоциональных оценок и стремиться к приятным оценкам. Средством достижения этой цели должно быть только инициирование актуаторов, которыми

располагает организм. Эти актуаторы должны воздействовать не непосредственно на аппарат эмоций (в отличие от известной экспериментальной крысы, получившей возможность нажимать на педаль и возбуждать свой центр приятных ощущений через вживленный в него электрод, чем она и занималась до смерти), но на окружающую среду, обеспечивая возможные способы взаимодействия данного организма со средой. И только вызвав в среде появление таких объектов, процессов или явлений, воздействующих на сенсоры системы, которые будут распознаны этой же системой как образы, вместе с их атрибутами — определенными эмоциональными оценками, нервная система сможет тем самым повлиять на эмоциональную оценку текущего состояния. Тем самым, природа заставила организм, стремящийся лишь к получению приятных ощущений и к избежанию неприятных, взаимодействовать со средой, что приводит к накоплению новых знаний о свойствах системы «среда–организм» и повышает шансы организма на выживание. Если отключить аппарат эмоций, то организм перестанет что-либо делать, поскольку у него полностью пропадет мотивация к активности, и он погибнет в полном безразличии.

Задача установления правильного отношения эмоциональных оценок образов и объективной полезности для организма соответствующих им реальных явлений решается в природе естественным отбором. Успешнее выживают организмы, у которых образы получают более адекватные оценки, а при неправильных оценках, организм стремится к вредным воздействиям и погибает.

Описанный механизм побуждения к активности через аппарат эмоций является универсальным и в целом не зависит от конкретного содержания образов. Обычно при рассмотрении функциональных систем говорят о наличии определенных «мотиваций» как о стремлении удовлетворить определенные «потребности», которые тут же перечисляются, например, потребность к сытости, к размножению и еще несколько других. Однако описанное нами стремление к получению положительных эмоциональных оценок и к избежанию отрицательных оценок является более общим. Так, неважно, какой образ вызвал резко отрицательную оценку — от срабатывания рецепторов в пустом желудке или от падения курса акций на финансовой бирже, нервная система будет стараться инициировать действия, направленные на вытеснение неприятных ей эмоциональных оценок, а единственным способом для нее является совершение воздействий на среду и такое изменение ее состояния, которое вызовет вытеснение неприятных образов.

В. Аппарат эмоций обеспечивает выработку эмоциональных оценок образов. Пусть сформирован некоторый новый образ O_j . Эмоциональную оценку $p_{z,j}$ для этого образа нервная система может формировать в процессе одновременного наблюдения за эмоциональной оценкой текущего состояния системы $S(t)$ в моменты, когда данный образ $O_j(t) = 1$ распознан, или в некоторых окрестностях этих моментов времени. Если каждый раз при распознавании данного образа оценка $S(t)$ падает, это является основанием для приписывания этому образу низкой оценки, и, соответственно, наоборот. Конкретно, оценка $p_{z,j}$ образа O_j может быть равна среднему арифметическому от оценок $S(t)$ в моменты, когда данный образ распознан, т. е. когда $O_j(t) = 1$. Практически, сразу после формирования образа его оценка нейтральна, но с накоплением статистики она может измениться.

Оценка каждого вновь сформированного образа в свою очередь начинает участвовать в выработке эмоциональной оценки текущего состояния системы $S(t)$, поэтому наше состояние начинает определяться и вкладом новых сформированных нами образов. Этот итерационный процесс формирования оценок образов начинается с оценок образов, изначально сформированных в организме, т. е. переданных ему по наследству от предков данного вида, и приспособленных к усредненным условиям существования организмов данного вида. Поскольку идея определения оценок качества образов также основана на корреляционном анализе сигналов от распознанных образов и сигналов оценки S^t текущего состояния ОУ, то эту задачу также можно решать с помощью сети из указанных нейронов.

С. Существует специальная шкала и средства для выражения эмоциональных оценок. Эмоциональные оценки как отдельных образов, так и общего текущего состояния носят сравнительно объективный характер и могут существенно отличаться для разных организмов даже одного вида. Такие оценки не слишком удобны для принятия решений, особенно при общении с себе подобными. Поэтому можно наблюдать существование дополнительной шкалы оценок, используемых, по-видимому, как для непосредственного управления, так и при общении между организмами. Назовем такого рода оценки выраженными эмоциональными оценками и будем обозначать такие оценки символом $B(t)$.

Формально упорядоченная шкала оценок $B(t)$ подобна шкале оценок $S(t)$, с той разницей, что она содержит меньше оценок, т. е. они более грубы, однако имеют (в первую очередь у людей) уже словесное выраже-

ние. Так, можно весьма правдоподобно определить эту шкалу следующим множеством оценок:

$$B = \{b_1, b_2, \dots, b_h, \dots, b_H\},$$

где b_1 = «невыносимо плохо», b_2 = «очень плохо», b_3 = «плохо», b_4 = «так себе», b_5 = «хорошо», b_6 = «очень хорошо», b_7 = «прекрасно», b_8 = «недо-стижимо прекрасно». В каждый текущий момент t инициирована только одна из оценок $B(t) = b_h \in B$.

Существует некоторое пропорциональное отображение ζ из множества S в множество B , $\zeta : S \rightarrow B$ такое, что всегда текущая оценка $S(t)$ отображается в текущую оценку $B(t)$. Однако не все множество S может быть испытано нервной системой данного индивида к текущему моменту, но только оценки из диапазона $[S_{min}, S_{max}]$. Поэтому отображение ζ переводит S_{min} в b_1 , а S_{max} в b_2 , что дает индивиду возможность пользоваться всем диапазоном оценок B . Такая шкала объясняет некоторые эффекты различного индивидуального восприятия явлений. Например, пусть индивид всю жизнь находился в очень благоприятных условиях и узнал только диапазон оценок $[S_{min1}, S_{max1}]$, а другой индивид жил в плохих условиях и узнал только оценки из диапазона $[S_{min2}, S_{max2}]$, пусть $S_{min2} < S_{min1}$ и $S_{max2} < S_{max1}$. Тогда один и тот же образ с оценкой $p_{z,j}$ может оцениваться первым индивидом, например, оценкой b = «плохо», а вторым индивидом — оценкой «очень хорошо». Другой эффект — если у некоторого индивида вдруг упала оценка S_{min} , то благодаря отображению ζ произойдет переоценивание образов и оценки всех образов автоматически возрастут: то, что оценивалось как плохое, станет хорошим, в том числе и оценка текущего состояния индивида.

Д. Аппарат эмоций играет большую роль в принятии управляющих решений. Алгоритм принятия решений, основанный на эмоциональных оценках, описан нами в [11]. Его идея состоит в следующем. В каждый текущий момент подсистема распознавания образов сообщает нервной системе, какие образы распознаны ею в текущий момент. Тем самым в базе знаний управляющей системы отделяются те ее области, которые адекватны данной ситуации, т. е. выделяется множество действий, которые вообще можно совершить в данной ситуации, согласно знаниям, накопленным нервной системой. В этих отделенных областях базы знаний нервной системой отыскиваются действия, которые обещают привести систему «среда-организм» в состояния, имеющие максимальные из возможных в

данной ситуации эмоциональные оценки, либо в состоянии, когда прекратят распознаваться образы с максимально плохими оценками. Найдя такое действие, нервная система дает команду исполнителям совершить его. Этот способ принятия решений одновременно активен и реактивен. Например, при отсутствии всякого внешнего стимула животное может встать и отправиться на поиски чего-то приятного, воспоминание о чем хранится у него в памяти, а в текущей ситуации он не видит для этого препятствий. Если же на животное подействовал некий раздражающий фактор, оно начнет совершать действия, направленные на его прекращение.

Е. Аппарат эмоций как внутренние системные часы нервной системы. При принятии решений в текущей ситуации нервной системе необходимо просмотреть свою базу знаний и найти оптимальное решение. Однако просмотр базы знаний требует определенного времени, которым нервная система может не располагать в текущих условиях. Определение времени, отведенного нервной системе на принятие решений в текущей ситуации, есть также функция аппарата эмоций. Внутреннее время нервной системы выражается в количестве $k(t)$ образов, которое можно успеть проанализировать в базе знаний при принятии решений в текущих условиях. Это число определяется некоторой функцией $k(t) = f(S(t), dS/dt)$ от значений общей текущей эмоциональной оценки и ее производной. Эта функция монотонно возрастает от некоторых предельно допустимых минимальных значений как $S_{min}(t)$, так и $(dS/dt)_{min}$, т. е., чем хуже ситуация или чем быстрее она ухудшается, тем скорее надо принимать решение. При минимальных граничных значениях функция равна нулю, здесь нервная система «падает в обморок», так как не в состоянии успеть принять решение. При больших значениях аргументов функция $f(\cdot)$ имеет плато или спад, что индивидуально для индивида, и в целом определяет тип его темперамента. Итак, в каждый момент функция $f(\cdot)$ аппарата эмоций подсказывает нервной системе, сколько образов с максимальными по модулю оценками она может успеть просмотреть в базе знаний с целью прогнозирования их поведения при альтернативных вариантах действий, и определяет лучшее из возможных действий с учетом последствий только по этим образам. Остальные образы, на учет которых в прогнозировании не хватило времени, получают в результате выбранного действия случайные изменения.

Е. Аппарат эмоций как канал передачи качественной информации. Наконец, следует упомянуть о важной роли участия аппарата эмоций в общении индивидов. При необходимости передачи знаний от одного ин-

дива к другому, животному приходится передавать информацию как об образах некоторых объектов, так и о своих эмоциональных оценках этих образов. Поскольку даже в элементарном передаваемом знании должна содержаться информация и об образах условий, и об образах действий, и об образах ожидаемых результатов действия, то нужно передавать и сведения об эмоциональных оценках каждого из таких компонент. Каналом для передачи такой многообразной эмоциональной информации служит мимика животного, управляемая аппаратом эмоций, с помощью которой передается и распознается такая эмоциональная информация об условиях, действиях и об ожидаемых результатах действия. Описанные здесь принципы работы аппарата эмоций использовались нами при построении прикладных систем на основе метода ААУ. Отметим, что общепринятое на сегодняшний день определение, что эмоции животных определяются какой-либо актуальной потребностью и оценкой вероятности ее удовлетворения [31], нам представляются мало конструктивными.

Общая схема системы ААУ

В целом УС, построенную по методу ААУ, в упрощенном виде можно изобразить следующей схемой (рис. 9). Заметим, что данная схема допускает весьма простые решения. Так, в указанной тройке множеств

*⟨множество образов условия, множество образов действия,
множество образов результатов⟩*

каждое из распознаваемых множеств может быть представлено, например, всего одним образом. Даже такое простое решение позволяет строить практически полезные приложения. Круг приложений систем ААУ будет расширяться по мере нахождения способов сокращения необходимого избытка нейронов. Пути решения проблемы избытка нейронов мы видим в:

- разработке методики задания оптимальной топологии нейронных подсетей УС на основе использования априорной информации об объекте управления;
- использовании генетических алгоритмов для выращивания сетей УС;
- применении известных методов поиска для динамического построения и перестроения сетей УС, а также использования элементов других технологий.

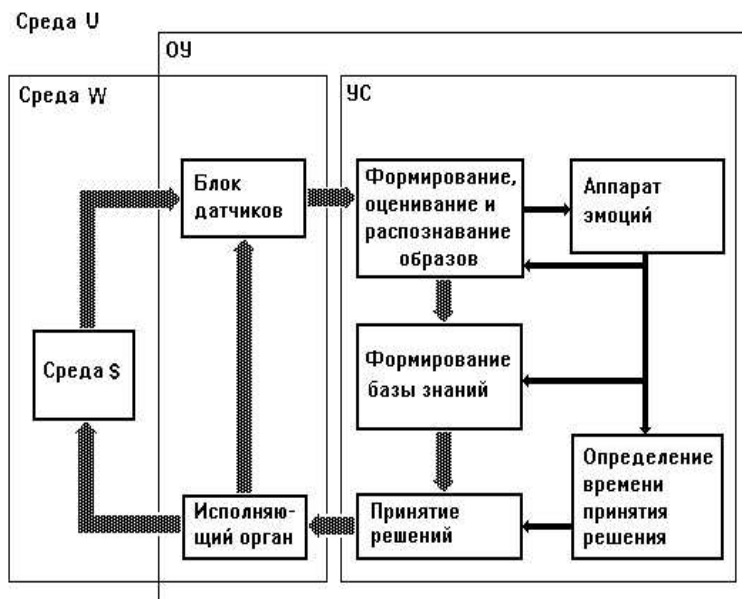


Рис. 9. Упрощенная схема управляемой системы, соответствующая методу автономного адаптивного управления (ААУ)

На основе метода ААУ можно строить разнообразные прикладные системы адаптивного управления. Так, нами были разработаны программные прототипы следующих систем.

1. Адаптивная система управления мобильным роботом Гном № 8, целевая функция которой состояла в выработке стереотипов поведения при обходе препятствий [34, 37].
2. Адаптивная система управления автомобильной подвеской (система AdCAS) [32].
3. Адаптивная система управления угловым движением космического аппарата (система «Пилот») [16, 19, 38].
4. Адаптивная система поддержки принятия решений (система «Тактик») [19].

Все системы показали способность адаптироваться к свойствам предъявленного объекта управления.

Сравнительный анализ метода автономного адаптивного управления и альтернативных подходов

Приведенное выше краткое описание метода ААУ позволяет определить круг альтернативных подходов, с которыми следует сравнить предлагаемый метод. Однако прежде выскажем несколько *общих положений*.

- Всякий подход имеет свою историю, он был задуман с некоторой определенной целью, нашел удачное практическое применение в некотором определенном направлении, имел какие-то более или менее продвинутые и удачные варианты и модификации, приспособляющие его для решения других задач. Очень часто исходные идеи, содержание теоретической проработки, круг наиболее широко используемых реализаций и инновационные модификации существенно различны. Поэтому при сравнении методов надо иметь в виду все эти аспекты.
- Все подходы, которые будут упомянуты ниже, некогда имели свое начало в задаче имитации живого мозга. Затем эти подходы сильно разошлись под влиянием их успехов в соответствующих им практических приложениях, более того, каждое из них породило свой спектр вариантов направлений. Сегодня это расхождение достигло таких степеней, что часто абсолютно утрачено общее понимание постановок задач, методов решения, языков формального описания, критериев оценки получаемых результатов и понимания перспектив. Однако нам представляется, что в множестве спектров направлений, на которые распался каждый из подходов, возможно пересечение, связанное с возвращением к исходной задаче имитации живого мозга. Те исследователи из разных направлений, которые озабочены целью понять и смоделировать принцип действия мозга, в отдаленной перспективе придут, хотя и своими путями, но к схожим решениям, поскольку имеют в виду один и тот же объект моделирования — мозг. В настоящее же время такие тенденции можно только с большим трудом угадывать.
- Развивая свой подход, мы не стремились изобрести новый способ управления, но шли от попыток понять способ управления, суще-

ствующий в природе. Главным критерием в теоретическом осмыслении для нас являлось соответствие биологическим оригиналам, тем их свойствам, которые понимались нами как наиболее важные. Естественно, что в практических приложениях приходится часто опираться на прагматические решения, при этом при необходимости мы упрощали идеи метода ААУ, но не меняли их на другие искусственные подходы, не соответствующие методу.

Метод ААУ и нейросети

Одним из возможных способов реализации управляющей системы, построенной в соответствии с методологией ААУ, является нейросетевая реализация на основе специально разработанных для данной системы нейроноподобных элементов и сетей. Тем самым появляются основания для сравнения системы ААУ с искусственными нейросетями (ИНС).

В своем традиционном виде ИНС были задуманы для решения задачи обучения распознаванию с учителем, либо аппроксимации функций. С этими целями они и используются в настоящее время наиболее широко. Назначение же системы ААУ — *автономное (без учителя) адаптивное управление*. Задача обучения распознаванию является только одной из подзадач управляющей системы ААУ. В последнее время в теории ИНС говорят о необходимости перехода ИНС к «управляющей парадигме», что, собственно, и реализовано в методе ААУ, если ее построить на нейросетях.

ИНС можно использовать тогда, когда имеется обучающая выборка, состоящая из двух множеств, одно из которых содержит векторы входных данных, а другое — соответствующие им векторы выходных данных. Если ИНС использовать как управляющую систему, то входные векторы можно интерпретировать как *образы ситуаций*, а выходные векторы можно интерпретировать как *коды действий* (стратегий), которые необходимо совершать при распознавании соответствующих ситуаций. Соответствие входов (образов ситуаций) и выходов (действий) для обучающей выборки должно быть известно заранее. Кроме того, над собственно нейросетью должно иметь место внешнее устройство, настраивающее веса связей нейронов. Систему ААУ можно использовать для управления в случаях, когда:

- закономерности во входной информации (образы ситуаций) заранее неизвестны и их следует найти;

- даже если образы будут обнаружены, то неизвестно, какие действия следует совершать при их распознавании;
- в управляющей системе имеется аппарат оценивания качества состояний объекта управления и целевые функции.

Например, если для некоторого робота заранее известно, какие действия следует совершать при наблюдении определенных классов ситуаций, то ИНС предварительно можно обучить давать команду на выполнение этих действий при распознавании класса ситуации. Систему ААУ целесообразно применять, когда заранее неизвестны ни классы ситуаций, ни оптимальные способы поведения при их наблюдении. Система ААУ могла бы делать следующее: самостоятельно формировать классы (образы) наблюдаемых ситуаций, оценивать степень качества (опасности, полезности) этих образов, находить адекватные способы воздействия на прообразы этих образов, находить оптимальные способы реагирования на распознаваемые ситуации и все это осуществлять в режиме реального управления. Поскольку качество управления в системе ААУ повышается по мере накопления ею знаний, а свойства объекта управления или среды могут меняться со временем, то возможно достижение состояний, когда будет получено более высокое качество управления, чем то, которое было зафиксировано в обучающей выборке для ИНС.

ИНС представляют собой, как правило, регулярные однородные сети с полносвязными соединениями нейронов соседних слоев. Управляющая система ААУ состоит из нескольких подсетей с разным функциональным назначением и различным характером связей. При этом отдельный нейрон может быть связан с нейронами из разных слоев. В методе ААУ отдельный нейрон связан с отдельным образом, в ИНС — нет. Нейроноподобные сети в системе ААУ являются, тем самым, *семантическими сетями*.

Формальные нейроны в методе ААУ существенно отличаются от формальных нейронов в ИНС. Нейрон в методе ААУ более сложен функционально. Он автоматически накапливает статистику в нескольких своих узлах и меняет свое функционирование при обнаружении определенных закономерностей. В нейронах ИНС этого нет. Нейрон ААУ имеет и учитывает временные задержки, что принципиально важно для обнаружения причинно-следственных связей в системе. Нейроны и сети ИНС непосредственных временных задержек не имеют.

Роль «учителя» в методе ААУ играет *аппарат эмоций* — важная многофункциональная подсистема, обеспечивающая внутреннюю активность,

целеполагание, оценку качества состояний системы и объектов знаний, которая работает параллельно с основным контуром обработки внешней входной информации. В теории ИНС только появляются предложения о введении в сеть дополнительной подсети, обеспечивающей целеполагание.

Видимо, наиболее близким аналогом системы ААУ в области ИНС являются сети Хопфилда [35], имеющие обратные связи. Однако даже обученная сеть Хопфилда способна только реагировать на предъявленную ситуацию, в то время как система ААУ имеет внутреннюю активность, заставляющую ее даже в благоприятных условиях совершать некоторые действия, направленные на поиск новых знаний. Это принципиально разный способ поведения.

Преимущество ИНС перед распознающей нейросетью системы ААУ состоит в удобстве перехода от континуальных величин к дискретным. Поэтому ИНС можно использовать, например, в роли блока датчиков для системы ААУ, а, возможно, и в других подсистемах.

Сближение подходов ИНС и ААУ будет происходить по мере развития ИНС в таких направлениях, как:

- переход к «управляющей парадигме»;
- дополнения ИНС, как распознающего блока, другими блоками, выполняющими получение и представление знаний, моделирование эмоций, принятие решений, т. е. блоками, в совокупности образующими автономную адаптивную управляющую систему;
- переход от обучения с учителем к самообучению;
- уход от однородности и полносвязности сети;
- внесение временного параметра в работу нейросетей;
- переход к формальным нейронам, более адекватным биологическим нервным клеткам в отношении их способности к автоматической самонастройке.

Со стороны системы ААУ сближение возможно при развитии в методе ААУ подходов к работе с континуальными параметрами, описывающими границы классов. Это может быть введение непрерывных весов в состав формальных нейронов ААУ, либо использование традиционных ИНС для реализации различных подсистем системы ААУ.

Метод ААУ и экспертные системы

При описании структуры управляющей системы ААУ в ней, при желании, можно выделить отдельно систему, которую можно назвать *базой знаний* (БЗ). К БЗ можно отнести совокупность нейронов, сформировавших такие образы, в прообразы которых входила информация о действиях, совершенных самой управляющей системой (см. выше, раздел «Имитационный метод автономного адаптивного управления»), т. е. прообразы, которые можно интерпретировать как импликацию

условие \rightarrow действие \rightarrow следствие.

Не все множество нейронов в управляющей системе обладает такими свойствами (отметим, что в мозге локализовать такую отдельную подсистему не удастся, она является распределенной). Если при анализе системы ААУ все-таки выделить БЗ как отдельную подсистему, то можно говорить, что система ААУ есть система автоматического получения, накопления и использования знаний в реальном времени управления. Тем самым появляются основания для сравнения системы ААУ с *экспертными системами* (ЭС) [36]. Различия удобно объяснить при сравнении данных направлений и систем по их целям, назначению, содержанию используемых в системе знаний, степени автоматизации и способам реализации.

Целью, стоящей перед разработчиками ЭС, является, по определению, создание прикладных интеллектуальных систем, предназначенных для оказания консультационной помощи специалистам, работающим в некоторых предметных областях. Целью методологии ААУ является построение концептуальной модели нервной системы и мозга, что относится к разряду фундаментальных проблем.

С точки зрения назначения систем, ЭС предназначена для представления в ЭВМ знаний, накопленных человеком-экспертом, для дальнейшего их тиражирования и использования в режиме поддержки принятия решений другими специалистами в данной предметной области.

Если знания, накопленные в мозге человека, ранжировать по их сложности, комплексности, абстрактности, то ЭС, по их первоначальному определению, предназначены для работы с вербализованными посредством человеческого языка знаниями из области высокоинтеллектуальной профессиональной деятельности человека-специалиста. В свою очередь, система ААУ оперирует с диапазоном знаний, начинающимся от элементарной ин-

формации, поступающей в двоичном виде от датчиков-рецепторов, и рассматривает процесс ее последующего агрегирования и усложнения.

Диапазон знаний, обрабатываемых системой ААУ, может расширяться до пересечения с диапазоном знаний, обрабатываемых ЭС, при соответствующем количественном расширении УС ААУ и использовании указанных возможностей формирования языка, вербализующего элементы знания. Система ААУ работает на своего рода «рефлекторном уровне», набирая статистику по прецедентам пространственно-временных вариаций наблюдаемой информации и их связям с выходными воздействиями, не вдаваясь в содержательный смысл найденных закономерностей, а только оценивая их влияние на целевую функцию. Видимо, здесь уместна аналогия с ситуацией в нервной системе ребенка, когда он, например, осваивает езду на велосипеде, находя закономерные связи пространственно-временных образов, оценивая их качество, и запоминая полученные знания в своей памяти, затем использует их для управления. Применение ЭС в диапазоне знаний, соответствующих процессам на уровне рецепторов и нейронов нервных систем, вряд ли целесообразно, так как выходит из сферы предназначения ЭС по их определению и неэффективно по реализации.

С точки зрения уровня автоматизации систем, система ААУ по своей идее — полностью автоматическая автономная система, работающая без участия человека, и реализующая в реальном времени процесс обучения и управления. ЭС по своей идее — средство автоматизации представления знаний, накопленных человеком-экспертом, и использования этих знаний человеком в режиме консультирования.

С точки зрения способов реализации, наиболее органичным для системы ААУ способом является нейросетевой подход в его аппаратном воплощении, наиболее адекватном строению естественных нервных систем. Подходы к реализации ЭС по способам «представления знаний» и «рассуждений» ориентированы на программные реализации.

Общие черты в системе ААУ и ЭС можно увидеть в наличии в составе обеих систем блоков распознавания образов, представления знаний, принятия решений, что характерно для многих автоматических или полуавтоматических управляющих систем и интеллектуальных систем.

Сближение подходов ЭС и ААУ будет происходить по мере появления в ЭС следующих свойств:

- способность работать в автоматическом режиме и в режиме реального времени;
- ориентация на работу с элементарной сенсорной информацией;

- переход к автономным способам использования, что, вообще говоря, входит в противоречие с исходным назначением ЭС.

Системы ААУ могут приближаться к свойствам ЭС, если отказаться в них от автоматического принципа действия и перевести на полуавтоматический режим работы в качестве системы поддержки принятия решений.

Метод ААУ и системы нечеткой логики

В последнее время широкое развитие получают управляющие системы на основе *нечеткой логики* (fuzzy logic), т. е. логики, в которой используются нечеткие квантификаторы. Системы принятия решений на этой основе используют *рассуждения* (т. е. способы получения заключений или новых знаний) с подобными квантификаторами с помощью специальных приемов, к которым относятся фазификация, функции принадлежности, принятие решений на их основе, дефазификация. Идея *систем нечеткой логики* состоит в том, что человек, принимая решения, пользуется преимущественно нечеткими правилами. Например, человек думает примерно так: «Эта вода скорее горячая, чем холодная, поэтому я поверну кран немного влево». Попытка формализовать такого рода способ рассуждений, и воспроизводить его в компьютерной программе, и привела к созданию систем нечеткой логики. Таким образом, системы нечеткой логики предназначены для того, чтобы зафиксировать правила принятия решений, которыми пользуется эксперт, и воспроизводить их с помощью компьютера. Нечеткие правила принятия решений могут использоваться в экспертных системах.

Системы нечеткой логики обычно не предусматривают каких-либо автоматических процедур для порождения или коррекции правил принятия решений, т. е. в своем каноническом виде они не являются адаптивными. В рамках методологии ААУ были разработаны подходы к построению автоматических адаптивных систем управления на основе нечеткой логики, представленные в [24].

В целом, обоснованием использования нечеткой логики для построения систем управления является то, что человек в процессе принятия решений почти всегда пользуется именно неполной и недостоверной информацией, и, по-видимому, это свойство мозга обусловлено свойствами самих биологических нейронов и их сетей.

Заметим, что как и в случае ИНС, системы нечеткой логики выделяют и эксплуатируют только одно из свойств мозга. Целью методологии ААУ является построение концептуальной модели нервной системы как целого.

Метод подкрепляющего обучения

Метод *подкрепляющего обучения* (reinforcement learning) позволяет управляющей системе автоматически создавать базу знаний, представляющую собой отображение из множества состояний, в которых может находиться объект управления, в множество качественных оценок, жестко связанных с положением целевого состояния, которого должен достичь объект управления. Например, если бы речь шла о мобильном роботе, передвигающемся в метрополитене к некоторой фиксированной заданной станции, то управляющая система на основе подкрепляющего обучения выработает такую базу знаний, которая будет представлять собой список станций с указаниями, в какую сторону ехать, если ты находишься на этой станции. Если же изменить положение цели, то управляющую систему следует переучивать.

В системе ААУ база знаний устанавливает отображение из множества состояний в это же множество состояний, плюс множество оценок, связанных с целевым состоянием или другой целевой функцией. Таким образом, в случае с метрополитеном, база знаний представляла бы собой карту метрополитена, пользуясь которой, можно было бы передвигаться по метро к любой из станций, а также прокладывать многошаговые маршруты. Поэтому при изменении целевой функции управляющей системе не нужно было бы полностью переучиваться.

Некоторые завершающие замечания

Заключая данный раздел, необходимо отметить следующее. Как было показано выше, всякая управляющая система находится в цикле управляемого взаимодействия с объектом управления и средой (см. рис. 9). Управляющая система, которая претендует на имитацию мозга, должна состоять из определенных подсистем — аппарата формирования и распознавания образов, базы знаний, аппарата эмоций, аппарата принятия решений, которые принадлежат названному циклу прохождения информации. Очевидно, что в разных приложениях в зависимости от условий, имеются возможности сильного упрощения отдельных подсистем такой полной управляющей системы или их замены «заглушками», при этом сам цикл управляемого взаимодействия может сохраняться. Соответственно, центральная задача построения автономной адаптивной системы управления в общем виде существенно изменяется и возникают *частные постановки задач* и планы их решений.

1. Если априори достоверно известно, как следует управлять в той или иной ситуации, а вся проблема состоит только в правильном распознавании ситуации, то УС можно построить на основе системы распознавания, разработка которой и потребует основных усилий. Роль остальных подсистем в УС (рис. 9) становится элементарной. Здесь для построения УС потребуются знания теории распознавания образов. Если при этом задача распознавания плохо формализуется, но имеется обучающая выборка, то можно использовать нейросети. Бытует мнение, что вся проблема управления может быть сведена к проблеме распознавания.
2. Если для управления требуется только использовать удобным образом представленные знания, накопленные ранее, а все остальные подзадачи решены и данные определены, то в составе УС на первое место выдвигается база знаний, а остальные подсистемы вырождаются. Проблема сводится к способам представления знаний, автоматизации вывода новых знаний из уже имеющихся, интерпретации полученных выводов и т. д. В этом случае УС можно построить на основе экспертной системы с соответствующим упором на решение проблем получения, представления и использования знаний.
3. Если ни задача распознавания, ни задача получения и представления знаний в управляющей системе не актуальны, т. е. все данные и решения для этих блоков УС определены, а проблема состоит только в принятии оптимальных решений на основе текущей информации и имеющихся знаний, то проблема построения УС превращается в задачу выбора и принятия решений с применением соответствующих подходов. В частности, при чрезмерной сложности строгих процедур принятия решений, возможно применение робастных подходов или систем нечеткой логики. Если зависимости не слишком сложны, но вид их известен, а надо только найти подходящие значения параметров, используются параметрические подходы к построению адаптивных систем управления, возможны и непараметрические случаи для зависимостей произвольного вида.
4. Если известны все данные и все возможные решения для всех подсистем УС, заданы некоторые критерии, определяющие требования к УС и ее качество, и ставится задача добиться наибольшего качества работы системы, то появляется необходимость в применении методов оптимизации системы.

Заметим, что названные здесь отдельные задачи к настоящему времени очень глубоко осмыслены и имеют свои решения. На основе этих решений строятся очень эффективные системы. Некоторые из предложенных решений не имеют никакого отношения к моделированию мозга и нервных систем, некоторые имеют, но относятся к так называемому «*программно-прагматическому*» направлению, моделирующему только конечный результат (например, распознающие системы, экспертные системы, системы нечеткой логики), некоторые относятся к «*имитационному*» направлению, моделирующему и результат и принцип решения (например, нейросети). Однако даже системы, моделирующие деятельность мозга, вычлениют только какое-то одно его частное свойство, хотя моделирование этого свойства может оказаться настолько полезным, что на его основе удастся строить действующие технические системы, эффективно решающие некоторый соответствующий им класс задач. Можно утверждать, что для любого подхода такого рода всегда найдется приложение, в котором этот подход даст оптимальное решение. Сравнение таких решений даже с решениями мозга человека часто оказывается не в пользу последнего. Например, автопилот способен управлять самолетом более качественно, чем летчик, но только в отдельных режимах. Система принятия решений может лучше человека играть в шахматы, но решать только эту задачу. Экспертная система может обладать знаниями в некоторой предметной области, которые превосходят знания любого специалиста. Система логического вывода может лучше человека доказывать теоремы. Но каждая из таких систем по принципу действия, а также по своим функциям не соответствует мозгу как таковому. Вопрос о моделировании мозга остается открытым. Целью методологии ААУ является построение именно *концептуальной модели нервной системы*.

В каждом из этих частных решений, подменяющих задачу построения автономной адаптивной системы управления в общем виде задачей построения только одной из ее подсистем, присутствуют определенные исходные данные, заслоняющие и подменяющие собой все другие подсистемы. Теперь представим ситуацию, когда не определены или плохо определены данные для всех подсистем УС. Например, УС «видит» входные данные, но не видит в них никаких закономерностей, а если какие-то закономерности обнаружила, то не понимает, хорошо это для нее или плохо, не видит разницы между вариантами действий, выполнение которых она может инициировать, не видит связи между своими действиями и поступающей на вход информацией, а если видит, то не умеет использовать, а если умеет,

то не понимает с какой целью, а если понимает, то не всегда успевает, и т. д. (читатель может проанализировать свой собственный опыт, когда он оказывался в незнакомых условиях, что типично для всех биологических систем). Анализ именно этой ситуации, а также поиск решений и есть предмет исследования методологии ААУ. По-видимому, эта задача комплексного построения системы, работающей в названных условиях, отличается от задач, перечисленных выше в пунктах 1–4 и не сводится к ним. Как минимум, речь идет о поиске взаимосогласованного решения всего комплекса названных задач. Отсюда можно сделать вывод, что вопрос о сравнении системы ААУ и перечисленных выше «альтернативных» систем аналогичен попытке сравнить некоторую цельную систему, собранную из подсистем, с каждой из таких подсистем, взятых по-отдельности.

В завершение необходимо сказать, что при постановке задачи построения системы автономного адаптивного управления в общем виде имеется *проблема философского плана*. Нужны ли такие управляющие системы, где может понадобиться искусственная нервная система? В научно-техническом развитии набрана большая инерция разработки и использования именно частных случаев управляющих систем. От автоматических устройств требуется только быть точными исполнителями строго определенных заданий. Нужна ли кому-нибудь саморазвивающаяся, имеющая собственное целеполагание и внутреннюю активность, совершающая пробные поисковые движения, обладающая собственными эмоциями, зависящая от случайностей в своей предыстории управляющая система? Чтобы оценить возможности практического использования искомой совершенной модели какой-либо нервной системы в случае ее создания (например, нервной системы мышцы), достаточно представить себе возможности практического использования оригинала — живой мышцы. Поиск сферы практического использования таких систем в их полном виде нетривиален.

Системы, обладающие свойствами рассмотренных систем ААУ, не соответствуют общепринятому пониманию и назначению систем искусственного интеллекта (ИИ), к которым сегодня относят экспертные системы, распознающие системы, нейросети, лингвистические процессоры и робототехнику. Поэтому предлагается выделить системы с подобными свойствами в отдельное направление «*Системы автономного искусственного интеллекта*» (АИИ) [13]. Если основное назначение систем ИИ — быть исполнителями заданий, поступивших извне от пользователя, то назначение систем АИИ — выживать и накапливать знания, подчиняясь более свободному от внешних управляющих воздействий внутреннему целеполаганию.

Помимо теоретического интереса, прагматическое значение систем АИИ, имитирующих естественные организмы, может состоять в их способности к выживанию, работе и накоплению знаний в плохоформализуемых условиях, требующих адаптации и саморазвития.

Заключение

В настоящей лекции с минимальным математическим формализмом был изложен возможный подход к решению проблемы построения *имитационной модели нервной системы*, названный методологией автономного адаптивного управления (ААУ). Прагматическое значение рассмотренного подхода состоит в том, что он позволяет строить действующие, хотя и со сравнительно небольшим пока числом ($\approx 10^4 - 10^5$) образов, но практически полезные быстродействующие адаптивные управляющие системы, пригодные для управления объектами, плохо поддающимися формализации. Приведены примеры простых прототипов практических приложений систем. Показано, что по мере увеличения возможного числа эмулируемых нейронов, а также по мере решения проблем перехода от континуальных величин к дискретным, «интеллект» системы ААУ может существенно возрастать. Проведен сравнительный анализ систем ААУ с альтернативными подходами к построению управляющих систем.

Мы полагаем также, что системы ААУ могут составить самостоятельное направление «системы автономного искусственного интеллекта (АИИ)», имеющее свое назначение и сферу применения [13].

Литература

1. Винер Н. Кибернетика: Пер. с англ. 2-е изд. – М., 1968.
2. Cannon W.B. Bodily changes in pain, hunger, fear and rage. – Appleton-Century, 1929.
3. Анохин П. К. Теория функциональной системы // *Тр. Международного симпозиума по техническим и биологическим проблемам управления* (Ереван, 24–28 сент. 1968 года). Общие вопросы физиологических механизмов. Анализ и моделирование биологических систем. – М.: Наука, 1970.
4. Ляпунов А. А., Беликова М. А. О кибернетических вопросах биологии // В кн.: *О некоторых вопросах кодирования и передачи информации в управляющих системах живой природы*. – Новосибирск, 1971.

5. Mesarovic M.D. Conceptual basic for a mathematical theory of general systems // *Kybernetes*, **1**, 1972.
6. McCulloch W.S., Pitts W.A. logical calculus of the ideas immanent in nervous activity // *Bull. Math. Biophys.* – 1943, **5** [Рус. пер.: Маккалок У., Питтс У. Логическое исчисление идей, относящееся к нервной активности // В сб. «Автоматы». – М.: ИЛ, 1956]
7. Розенблат Ф. Принципы нейродинамики. – М.: Мир, 1965.
8. Жданов А. А. О подходе к моделированию управляемых объектов. – Препринт ВЦ РАН СССР. – М., 1991.
9. Zhdanov A. A. Application of pattern recognition procedure to the acquisition and use of data in control // *Pattern Recognition and Image Analysis*. – v. 2, No. 2, 1992.
10. Zhdanov A. A. A principle of pattern formation and recognition // *Pattern Recognition and Image Analysis*. – v. 2, No. 3, 1992.
11. Жданов А. А. Об одном имитационном подходе к адаптивному управлению // Сб. «Вопросы кибернетики». Научный совет по комплексной проблеме «Кибернетика» РАН. Вып. 2. – М., 1996.
12. Жданов А. А. Формальная модель нейрона и нейросети в методологии автономного адаптивного управления // Сб. «Вопросы кибернетики». Научный совет по комплексной проблеме «Кибернетика» РАН. Вып. 3. – М., 1997.
13. Жданов А. А. О понятии автономного искусственного интеллекта // Сб. научн. тр. «Искусственный интеллект в технических системах». – М.: ГосИФТП, 1997.
14. Zhdanov A. A. About an autonomous adaptive control methodology // *ISIC/CIRA/ (ISAS'98)*, NIST, Gaithersburg, Maryland. September 14–17, 1998.
15. Zhdanov A. A. The mathematical models of neuron and neural network in autonomous adaptive control methodology // *WCCI'98(IJCNN'98)*, *IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, May 4–9, 1998.
16. Жданов А. А., Норкин Н. А., Гуриев М. А. Некоторые практические приложения метода автономного адаптивного управления // Сб. научн. тр. «Искусственный интеллект в технических системах». Вып. 19. – М.: ГосИФТП, 1998.
17. Жданов А. А., Винокуров А. Н. Система проектирования и исследования нейронных сетей «СПИНС» в методологии автономного адаптивного управления // Сб. научн. тр. Всероссийской научно-техн. конференции «Нейроинформатика-99», ч. 1. – М.: МИФИ, 1999.
18. Рябовиков А. В., Жданов А. А. О некоторых формальных моделях нейронов // Сб. научн. тр. Всероссийской научно-техн. конференции «Нейроинформатика-99», ч. 1. – М.: МИФИ, 1999.

19. Жданов А. А., Арсеньев С. В. О некоторых приложениях принципа автономного адаптивного управления // Сб. научн. тр. Всероссийской научно-техн. конференции «Нейроинформатика-99», ч. 3. – М.: МИФИ, 1999.
20. Жданов А. А., Винокуров А. Н. О нейросетевом методе «Автономного адаптивного управления» // Сб. докл. V Всероссийской конференции «Нейрокомпьютеры и их применение (НКП-99)». – М., 17–19 февраля 1999.
21. Zhdanov A. A., Vinokurov A. N. Emotions simulation in methodology of autonomous adaptive control // *14th IEEE International Symposium on Intelligent Control / Intelligent Systems and Semiotics ISIC/ISAS'99*. Special session *Emotions and Intelligent Systems*. September 15–17, 1999. – Cambridge, Massachusetts, USA. Paper No. 99-0021-6.
22. Жданов А. А. Об одном имитационном подходе к адаптивному управлению // Сборник «Вопросы кибернетики», № 2. Научный совет по комплексной проблеме «Кибернетика» РАН. – М., 1996. – С. 171–206.
23. Жданов А. А. Метод автономного адаптивного управления // *Известия Академии Наук. Теория и системы управления*. – 1999, № 5. – с. 127–134.
24. Жданов А. А., Караваев М. В. Применение нечеткой логики в имитационной системе автономного адаптивного управления // *Труды института системного программирования*. Под ред. В. П. Иванникова. – М.: Изд-во МГУ, 2002. – С. 119–135.
25. Zhdanov A. A., Ryadovikov A. V. Neuron models in the autonomous adaptive control method // *Optical Memory and Neural Network*. – 2002. – v. 9, No. 2. – pp. 115–132.
26. Жданов А. А., Устюжанин А. Е. Возможности использования технологии детерминированного хаоса в системах автономного адаптивного управления // *Тр. Ин-та системного программирования*. Том 2. – М.: ИСП РАН, 2001. – С. 141–179.
27. Zhdanov A. A., Zemskikh L. V. The evolutionary growth of neural networks for the autonomous adaptive control system // *The 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001)*, Orlando, USA, July 22–25, 2001. – pp. 404–409.
28. Жданов А. А., Винокуров А. Н. Нейросетевой метод автономного адаптивного управления // «Нейрокомпьютеры: Разработка, применение». – № 2, 2001.
29. Жданов А. А., Одинцов К. В. Возможности представления измерительной информации с помощью формальных нейронов // Сб. научн. тр. «Искусственный интеллект в технических системах». Вып. 20. – М.: ГосИФТП. – 1999. – с. 111–146.

30. Жданов А. А. Моделирование высшей нервной деятельности // *Наука и жизнь*. – 2000, № 1, с. 58–64; № 2, с. 14–16.
31. Психофизиология / Под. ред Ю. И. Александрова. – Изд-е 2-е. М.–СПб и др.: Питер, 2003. – с. 267.
32. URL: <http://www.ispras.ru/~zhdanov/>
33. Чечкин А. В. Математическая информатика. – М.: Наука, 1991.
34. Жданов А. А., Крыжановский М. В. Преображенский Н. Б. Нейронная адаптивная система управления // Труды Международной конференции «Интеллектуальные и многопроцессорные системы» (IMS'2002), Кацивели. – С. 115–118.
35. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities // *Proc. of the National Academy of Science*. – 1982. – 79. – pp. 2554–2558.
36. Bernold T. Expert systems and knowledge engineering. – Elsevier, Science Publishers, 1986.
37. Жданов А. А., Крыжановский М. В., Преображенский Н. Б. Бионическая интеллектуальная автономная адаптивная система управления мобильным роботом // *Мехатроника*. – 2004. – № 1 (Принята редакцией).
38. Жданов А. А., Земских Л. В., Беляев Б. Б. Система стабилизации углового движения космического аппарата на основе нейроноподобной системы автономного адаптивного управления // *Космические исследования*, 2004 (Принята редакцией).

Александр Аркадьевич ЖДАНОВ, доктор физико-математических наук, заведующий отделом Института системного программирования РАН, Москва. Области научных интересов: адаптивное управление, нейронные сети, распознавание образов. Имеет 60 научных публикаций.

Л. А. СТАНКЕВИЧ

Санкт-Петербургский государственный политехнический университет

E-mail: stank@phtf.stu.neva.ru

НЕЙРОЛОГИЧЕСКИЕ СРЕДСТВА СИСТЕМ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РОБОТОВ

Аннотация

В данной работе обсуждаются возможности использования в системах управления интеллектуальных роботов нейрологических средств управления (базовых модулей, когнитивных и актуаторных структур), способных обучаться и формировать рациональное поведение. Показано, что системы управления современными интеллектуальными роботами по функциям и организации должны быть подобны нервной системе человека. Предлагается создавать такие системы с использованием нейрологических средств, техники слияния информации, техники интеллектуального управления, а также многоагентной технологии. Приводится пример разработки и применения нейрологических средств в системе управления гуманоидного робота.

LEV A. STANKEVICH

Saint-Petersburg State Polytechnical University

E-mail: stank@phtf.stu.neva.ru

NEUROLOGICAL MEANS FOR CONTROL SYSTEMS OF INTELLECTUAL ROBOTS

Abstract

Usage possibilities are discussed for neurological control means (basic modules, cognitive and actuator structures) as applied to control systems of intellectual robots. These means are suggested to be able to learn and to form a rational behavior. It is shown control systems for the modern intellectual robots must be similar functionally and structurally to the human nervous system. Such systems could be developed using neurological means, sensor fusion and intelligent control techniques as well as multiagent technology. An example of development and application of the neurological means with reference to control system for humanoid robot is considered.

Введение

В последнее время обсуждаются возможности использования в системах интеллектуального управления специальных *нейрологических средств*, способных обучаться в реальном времени отображению сложных функций и процессов [1, 2]. Эти средства имеют в основе нейронные сети и логические системы на правилах. Базовыми нейрологическими модулями можно считать клетки и ядра, реализующие аппроксимацию функций. Эти модули могут иметь различные базисы: интервально-логический, нечетко-логический и вероятностно-логический. Модули могут использоваться как компоненты локальных сетей, способных преобразовывать образы, и распределенных сетей, способных реализовать сложные процессы управления, составленные из компонентных функций.

На базе нейрологических средств, можно строить *когнитивные и актуаторные структуры*, способные обучаться, формировать рациональное поведение и исполнять действия объектов в среде. Использование таких структур позволяет создавать *интеллектуальные системы управления* динамическими объектами со *сложным поведением*, составленным из множества более простых поведений, которые комбинируются нужным образом в зависимости от текущей ситуации.

Наиболее сложное поведение может быть реализовано при использовании наряду с когнитивными а актуаторными структурами, построенными на нейрологических средствах, *многоагентной технологии* [3]. В этом случае сложное поведение может быть декомпозировано на ряд менее сложных поведений, каждое из которых может быть реализовано в структуре внутреннего (возможно виртуального) агента. Многоагентная технология позволяет организовать оптимальное взаимодействие между такими агентами, в результате которого может генерироваться много более сложное поведение, чем то, которое могли бы обеспечить отдельные агенты.

Разработка *интеллектуальных роботов* является одним из важных направлений, объединяющих искусственный интеллект и робототехнику. Интеллектуальные роботы прошли путь развития от исследовательских вариантов интегральных роботов [4] до современных роботов гуманоидного класса [5, 6], которые в ближайшем будущем должны стать полноценными помощниками человека, имеющими его образ и близкое к человеческому поведение.

Системы управления интеллектуальных роботов постоянно совершенствуются. Если для управления первыми интегральными роботами исполь-

зовались в основном эвристики и методы, основанные на использовании символьных знаний [7], то в дальнейшем стали широко применяться и *методы, основанные на нейронных и нейробиологических сетях* [8]. При создании систем управления современными интеллектуальными роботами значительное повышение эффективности управления может быть достигнуто применением нейробиологических средств.

Система интеллектуального управления при использовании нейробиологических средств, на которых строятся когнитивные и актуаторные структуры виртуальных агентов, становится похожей по функциям и организации на нервную систему человека. Поэтому ее можно условно назвать *искусственной нервной системой*. Целесообразно создавать такую систему, как иерархическую многосвязную систему, все уровни которой реализуются с использованием нейробиологических средств [9].

Примером использования нейробиологических средств является разработка *системы управления для антропоморфного робота АРНЭ* с 28-ю степенями подвижности [10]. Система интеллектуального управления этого робота построена с использованием нейробиологических средств и комбинированной технологии на основе техники слияния сенсорной информации и техники интеллектуального управления.

В перспективном проекте *системы интеллектуального управления гуманоидного робота* наряду с использованием нейробиологических средств, предполагается использование многоагентной технологии и средств самоорганизации. Это может позволить организовать поведение робота, подобное поведению человека.

Нейробиологические средства систем управления

Ранние исследования по нейрофизиологии позволили разработать теорию формальных нейронных сетей. Сейчас формальные нейронные сети успешно используются при решении задач, связанных с распознаванием образов и управлением, в том числе и в робототехнике. Исследования по нейрофизиологии показали также, каким образом некоторые функции и процессы реализуются в нейронных структурах мозга. Так, на основании исследования мозжечковых структур было показано, что информация на входах в нейронные структуры кодируется с использованием принципов грануляции, т. е. квантования, учитывающего неопределенности информации, и обрабатывается ассоциативно и логически [11]. Раскрыты также некоторые

методы формирования и ассоциативной обработки функций в нейронных сетях мозга. Это позволило разработать *нейрологические средства*, основанные на слиянии принципов обучения в нейронных сетях и принципов ассоциативной и логической обработки информации

Нейрологические средства позволяют организовать процессы решения задач в системах управления сложным поведением максимально подобно тем процессам обработки информации, которые имеют место в биологическом прототипе — центральной нервной системе человека. При этом предполагается, что имеет место [1, 2]:

- гранулированное представление информации, дающее возможность решать задачи в условиях неопределенностей;
- ассоциативная и логическая обработка когнитивных функций, позволяющая эффективно реализовать их отображение и сохранение;
- ассоциативная и логическая обработка когнитивных процессов, позволяющая производить параллельное и последовательное преобразование образов единообразно на всех уровнях системы управления.

Рассмотрим методы грануляции, ассоциативно-логической обработки функций и процессов, а также методы автоматического формирования таких функций и процессов.

Грануляция

Необходимость грануляции связана с неопределенностью информации в реальных условиях. Обработка информации в виде коллекций нечетко определенных гранул не является строго формализованной, но соответствует человеческой способности обрабатывать реальную информацию, получившуюся в результате грануляции объекта. Именно грануляция позволяет реализовать логические рассуждения и ассоциативную обработку реальной информации единообразно и подобно тому, как это делают структуры человеческого мозга.

Теория нечеткого информационного гранулирования разрабатывается Л. Заде и его последователями как обобщение нечеткого подхода [12]. Здесь обобщенная информационная грануляция (GIG — Generality Information Granulation) принята в качестве одной из базисных концепций когнитивной обработки информации. GIG предполагает, что любой составной информационный объект (переменная, отображение, образ) может быть декомпозирован на гранулы.

Каждая гранула является набором элементарных объектов, которые связаны вместе неопределенностью, близостью, подобностью и функциональностью. Формально объект O_c может быть представлен гранулированно, т. е.:

$$\begin{aligned} O_c &= \text{ins_}g(G_1, \dots, G_i, \dots, G_N), \\ G_i &= \text{has_}a(A_1, \dots, A_j, \dots, A_M), \\ G_i &= \text{has_}a(A_1, \dots, A_j, \dots, A_M), \\ A_j &= \text{has_}v(V_1, \dots, V_q, \dots, V_Q), \end{aligned} \quad (1)$$

где A_j — j -й атрибут гранулы G_i ; V_q — q -е значение атрибута A_j ; $\text{ins_}g$ — отношение включения для гранул; $\text{has_}a$ и $\text{has_}v$ — отношение «имеет» для атрибута и значения соответственно. Пример грануляции: объект «голова» включает гранулы «нос», «глаз», «щека» и т. д., имеющие атрибуты «размер», «форма» и т. п. со значениями, например, для первого атрибута — «большой», «малый» и т. п. Гранулы могут быть точными, т. е. с четкими границами (интервалы переменных, выделенные области определения функций и отношений, сегменты образов) и неточными — с нечеткими или вероятностными границами (терм-множества переменных, элементы нечетких или вероятностных графов, нечеткие или вероятностные правила и др.). Соответственно, атрибуты и значения точных и неточных гранул будут точными или неточными.

Ассоциативно-логическая обработка информации

Обработка гранулированной информации основана на использовании ассоциативных отображений (АМ). АМ является особой формой отображения векторной функции в виде сетевой вычислительной структуры, которая должна обладать хорошими аппроксимирующими свойствами. Возможно использование одиночных (SAM) и множественных (MAM) отображений.

Формально АМ, SAM и MAM могут быть представлены модельными наборами вида:

$$\begin{aligned} AM &= \{X, Y, S, U, F\}, \\ SAM &= \{P_x, P_y, AM\}, \\ MAM &= \{P_x, P_y, \{AM_t, t = 1, \dots, T\}\}. \end{aligned} \quad (2)$$

Компоненты, входящие в наборы, имеют следующие назначения: X, Y — множества входных и выходных параметров (векторные); S, U — множе-

ства структур и структурных единиц, их составляющих; F — множество базисных функций, реализуемых в узловых элементах; P_x, P_y — входной и выходной векторы; AM_t — SAM для t -го момента времени протекания когнитивного процесса; T — период времени процесса с фиксированными моментами t , в которые реализуются AM_t , составляющие M .

Предлагается реализовать ассоциативную обработку функций с использованием обучаемых сетевых компонентов типа клеток и ядер, а процессов — локальных и распределенных сетей, составленных из клеток и ядер.

Клетка (C — Cell) является минимальным обучаемым элементом, способным самостоятельно обрабатывать информацию в гранулированном виде. Формализованная модель клетки может быть представлена набором множеств:

$$M_c = \{X, W_c, H_x, S_c, H_y, BF, y\}, \quad (3)$$

где $X = \{x_0, \dots, x_n\}$ — множество входных параметров; $W_c = \{w_0, \dots, w_v\}$ — множество регулируемых весов ($m \geq n$); $H_x = \{h_{x1}, \dots, h_{xq}\}$ — множество скрытых входных параметров, соответствующих информационным гранулам на входах; $H_y = \{h_{y1}, \dots, h_{ye}\}$ — множество скрытых выходных параметров, соответствующих информационным гранулам на выходах; $S_c = \{s_{c1}, \dots, s_{cr}\}$ — множество связей скрытых входных и выходных информационных гранул, определяющих цепочку преобразований гранул при активизации этой связи; $BF = \{bf_1, \dots, bf_k\}$ — множество базисных функций активаторных элементов клетки; y — выходной параметр клетки.

Такая информационная модель поддерживается структурно-функциональной моделью клетки в виде (рис.1):

$$C = \text{ins}_f(IG, A(a_1, \dots, a_N, S_c, W_c), DG, T_c), \quad (4)$$

где IG — информационный гранулятор, формирующий множество H_x ; $A(\cdot)$ — активатор, состоящий из множества активаторных элементов $a_i, i = 1, \dots, N$, выполняющих преобразования скрытых информационных параметров и формировании множества H_y в соответствии со связями S_c и весами W_c ; DG — дегранулятор, формирующий выходной параметр y ; T_c — настройщик, формирующий множество связей S_c и весов W_c при настройке клетки на отображение $X \rightarrow y$, аппроксимирующее функцию $y = F(X)$; ins_f — отношение включения структурно-функциональных компонент.

Клетка является универсальным преобразователем информации (адаптивным аппроксиматором), имеющим n входов и один выход. Она соответ-

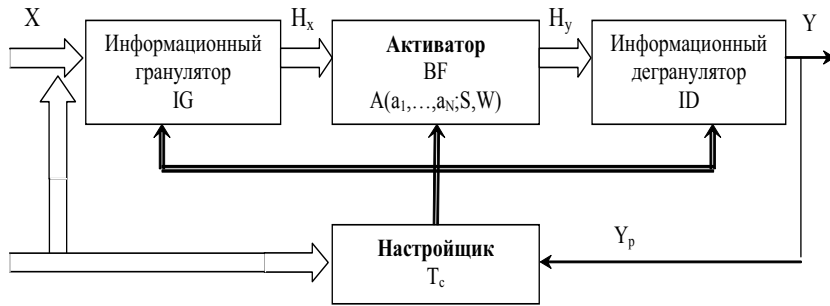


Рис. 1. Структурная модель клетки

ствует в модельном плане биологическому нейрону с его сетью синапсов, через которые организуются связи с другими нейронами, а также медиаторными регулирующими средствами. Преобразующие возможности клетки определяются набором базисных функций активаторных элементов и системой связей, образующих внутренние цепочки преобразований. Отображение $X \rightarrow y$ в клетке можно представить как цепочку отображений:

$$X \rightarrow y : (W_x, X) \rightarrow (Be, H_x) \rightarrow (W_h, S_c, BF_c) \rightarrow (Af, H_y) \rightarrow (w_y, y), \quad (5)$$

где множество входных параметров X представляется вектором с функционально определенными компонентами: (x_1, \dots, x_n) — сигнальные параметры (стимул клетки), x_0 — пороговый параметр; (W_x, X) — взвешивающее преобразование входного вектора X по вектору W_x ; (Be, H_x) — преобразование перед активизацией ($Be - Before$), в результате которого получается множество параметров H_x (грануляция и вычисление скрытых параметров: атрибутов и значений атрибутов); (W_h, S_c, BF_c) — внутреннее преобразование, регулируемое весами W_h и связями S_c и определяемое базисными активаторными функциями BF_c ; (Af, H_y) — преобразование после активизации ($Af - After$), в результате которого получается множество параметров H_y (дегрануляция и вычисление скрытых параметров). (w_y, y) — взвешивающее преобразование выходного параметра по вектору при его ветвлении и смешивании с выходными параметрами других клеток.

Заметим, что взвешивающие преобразования используют наборы регулируемых весов, входящие в множество $W_c = (W_x, W_h, w_y)$.

Клетка при обработке входной информации отображает ее в выходной

параметр с помощью базисных функций BF_c . Эти функции определяют вид преобразований гранул переменных, поступающих на входы активных элементов. В активаторе клетки активные элементы связаны в цепочки и образуют ассоциативную сеть, реализующую отображение информации от входов к выходу клетки. Полный набор базисных функций BF_c включает: логическое агрегатирование путем пересечения или объединения; инверсию и взвешивание в разных вариантах, определяемых типами переменных и гранул: четкие переменные и интервальные гранулы, нечеткие переменные и гранулы — нечеткие множества, вероятностные переменные и гранулы — вероятностные множества.

Ядро (N — Nucleus) содержит «склеенные» клетки (рис.2). Информационная модель ядра может быть формально представлена набором множеств (3), где $X = (X_1 \vee X_2 \vee X_M)$ представляет собой «склеенное» множество входов клеток, входящих в ядро; W_N , H_x , H_y — множества весов и скрытых параметров ядра, S_N — множество связей ядра, объединяющее связи клеток, BF — набор базисных функций активаторных элементов, одинаковый для всех клеток; скалярный выход y заменен на векторный Y — объединенное множество выходных параметров ядра.

Объединение клеток может быть полным, когда ядро имеет единый гранулятор, активатор, дегранулятор и настройщик, т. е.

$$N = \text{ins_}f(IG_N, A_N, DG_N, T_N). \quad (6)$$

В какой-то мере ядро с частичным объединением клеток можно считать однослойной сетью клеток с входами, параллельно подведенными к каждой клетке (от каждого входа к каждой клетке). Такое представление ядра используется в нейроинформатике для конструирования слоистых и модульных нейросетей. В данной работе оно использовано для моделирования биологических ядер, реализующих поведенческие функции с несколькими связанными параметрами, зависящими от ряда общих аргументов, т. е.

$$Y = F(X) \text{ или } y_j = f_j(x_1, \dots, x_N); j = 1, \dots, N. \quad (7)$$

Образ рассматривается как главный информационный элемент, который формируется как матрица или вектор со связанными координатами. Преобразование образов является основным операционным компонентом при реализации процессов, как последовательностей таких преобразований. Такие преобразования выполняются в структурных компонентах, таких как локальные или распределенные сети.

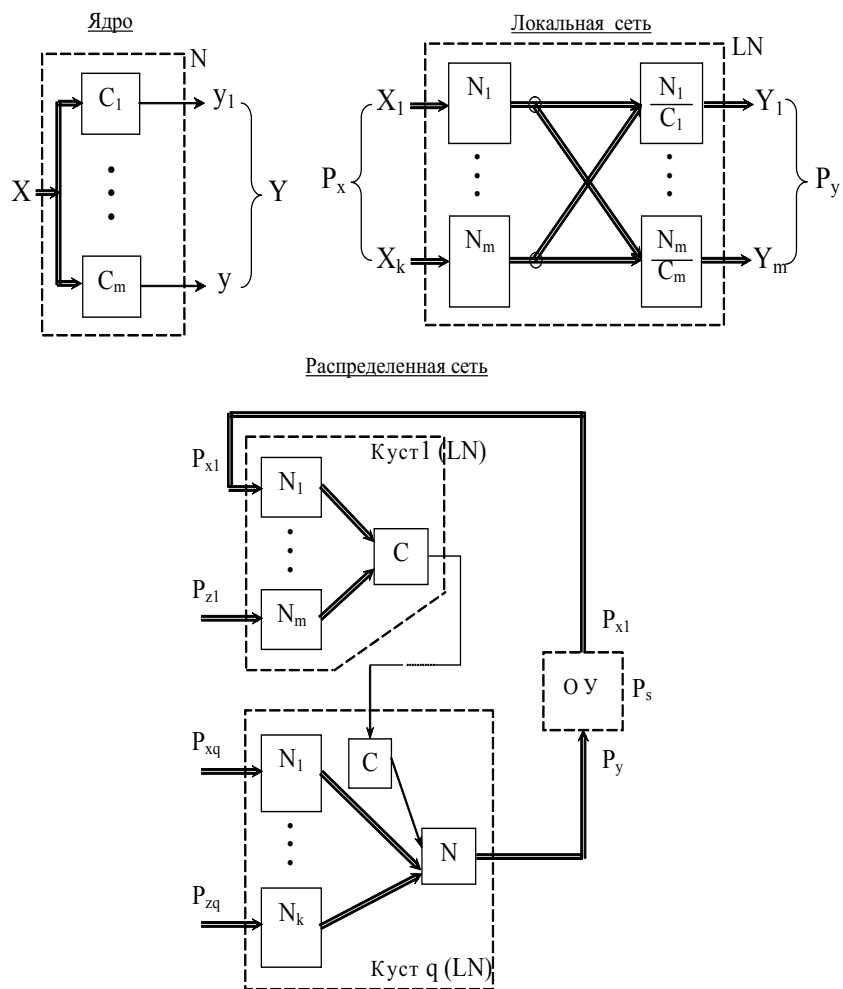


Рис. 2. Компонентные нейрологические средства

Локальная сеть (LN — Local Network) содержит связанные клетки и ядра. Локальная сеть рассматривается как самостоятельный модуль обработки образов (рис.2).

Информационная модель локальной сети может быть представлена набором множеств:

$$M = \{P_X(X, C_X), W_{LN}, S_{LN}, P_Y(Y, C_Y)\}, \quad (8)$$

где X и C_X — вектор образа P_X и связи между его координатами, т. е. множество входных параметров и связи между ними, определяющие входной (преобразуемый сетью) образ P_X ; W_{LN} — множество регулируемых весов связей между клетками и ядрами, входящими в локальную сеть; S_{LN} — множество связей клеток и ядер, входящих в локальную сеть; Y и C_Y — вектор образа P_Y и связи между его координатами, т. е. множество выходных параметров и связей между ними, определяющих выходной (преобразованный сетью) образ P_Y .

Структурная модель локальной сети имеет вид:

$$LN = (C_1, \dots, C_Q; N_1, \dots, N_K, L_{Ln}), \quad (9)$$

где C_i и N_j — i -я клетка и j -е ядро локальной сети, соответственно; L_{Ln} — обучатель (Learner), формирующий множество весов связей W_{LN} элементов сети.

Наличие разноуровневых компонентов (клеток и ядер) в локальной сети позволяет строить достаточно сложные преобразующие структуры с полносвязной и слоистой организацией. Локальные сети с полносвязной организацией используются для ассоциативного запоминания образов подобно описанным в литературе рекуррентным нейросетям с обратными связями. Локальные сети со слоистой организацией предназначены для классификации образов или аппроксимации заданных преобразований образов.

Распределенная сеть (DN — Distributed Network) имеет кустовую организацию. Они включают несколько подсетей (кустов), между которыми имеют место слабые (одинарные) связи (рис.2). Кусты строятся как локальные сети, обрабатывающие образы в соответствии с компонентными функциями процессов в системе. Так могут реализоваться процессы, определяющие сложное поведение объекта управления.

Распределенные сети могут преобразовывать образы замкнуто: образы, проходящие через кусты, возвращаются преобразованные опять на первый

куст. Динамический процесс преобразования образов в распределенной сети определяется цепочкой отображений:

$$P_{X/t_1} \rightarrow F_{k1}(P_X, P_{Z1}) \rightarrow \dots \rightarrow F_q(P_X, P_{Zq}) \rightarrow P_{Y/t_1} \rightarrow (OY) \rightarrow P_{S/t_2} \rightarrow P_{X/t_2}, \quad (10)$$

где t_1 и t_2 — моменты времени, соответствующие 1-му и 2-му циклам преобразований; F_{ki} — отображенные образы в i -ом кусте; P_S — образ состояния объекта управления; P_X , P_Y , P_Z — входной, выходной и внешний образы распределенной сети. При этом имеют место слабые связи между кустами, определяемые малыми весами $w_{i,i+1}$. Такая организация сетей позволяет учесть влияние на процесс управления дополнительных факторов, изменяющих это управление через вводимые в кусты внешние образы P_Z .

Нейрологические модули оказались удобными для адаптивной функциональной аппроксимации, т. е. формирования функций путем обучения в реальном времени, и кратковременного хранения при их частом использовании. Для долговременного хранения большого числа разных функций более удачным решением является «перекачка» информации в более компактные структуры нейронных сетей. Реализация процессов, составленных из последовательности функций, требует формирования соответствующих структур на цепочках нейрологических или нейронных компонент в решающей части системы, которая играет роль своеобразной рабочей памяти.

Автоматическое формирование функций, процессов. Представление функций в форме ассоциативных отображений предполагает использование для их формирования параметрического или структурного обучения. В первом случае при обучении изменяются только параметры фиксированных связей узловых элементов отображений, а во втором — сами связи и, значит, структура отображения.

Для разных вариантов обучаемых модулей, построенных для работы с функциями, могут быть использованы различные методы обучения и самообучения. Наилучшие результаты на практике дает применение модулей с нечетко-логическим базисом и локальным обучением. Однако такие модули расходуют много памяти и могут использоваться в основном для обучения в процессе работы и кратковременного хранения функций. Длительное хранение функций более выгодно с использованием модулей нейронного

типа, поскольку здесь требуется значительно меньше памяти. Однако время обучения в таких модулях значительно больше.

Представление процессов в виде цепочек функций также предполагает параметрическое и структурное обучение. Часто здесь может использоваться обучение с подкреплением (Reinforcement learning — RL) [13]. RL относится к целенаправленному обучению через взаимодействие со средой. Производится обучение тому, что делать, т. е. как отобразить ситуацию в действие, чтобы максимизировать числовой сигнал поощрения от среды. Две характеристики: поиск путем «проб и ошибок» и учет задержанного поощрения, отличают RL от других подходов. Любой метод, который пригоден для этой проблемы, может быть назван RL-методом. Полное определение проблемы RL в настоящее время дается в терминах оптимального управления и марковских процессов решений. В нашем случае обучение производится по эпизодам процесса, которые оцениваются специальной функцией с поощрением лучших эпизодов.

Топология структур системы управления, построенных из нейробиологических компонентов, может быть синтезирована обучением в соответствии с предварительно определенными целями управления в известной среде. По аналогии с нервной системой главным принципом формирования топологии является самоорганизация.

Самоорганизация является сложным процессом, включающим самосборку, самообучение и самосовершенствование системы. Самосборка может быть организована на основе процедур поиска в заданном топологическом пространстве возможных элементов структуры и связей между ними. Самообучение является оптимизационной процедурой с использованием в качестве критерия оптимизации некоторой функции цели. Такая процедура должна обеспечить настройку параметров структуры для того, чтобы обеспечить целенаправленное поведение объекта управления в непрерывно изменяемой среде. Самосовершенствование является процессом коррекции и дополнения начальной базы знаний о топологии структур системы управления (своеобразной генетической информации). Система может использовать эту информацию при самосборке и самообучении, чтобы направлять поисковые и оптимизационные процессы.

Процесс самоорганизации может быть реализован по-разному в зависимости от того, как должна быть организована система.

Централизованная организация предполагает, что система организуется одним нейробиологическим компонентом, содержащим информацию о структуре всей системы (метаинформацию) и обеспечивающим мониторинг те-

кущего качества системы в терминах цели.

Децентрализованная организация основана на том, что каждый нейрологический компонент должен содержать информацию об его соседях в структуре и получать информацию о текущем качестве всей системы. При этом каждый компонент сам определяет собственное поведение.

Комбинированная организация использует как централизованный, так и децентрализованный способы организации. Здесь все нейрологические компоненты имеют информацию о своих соседях в структуре и сами определяют свое поведение. Однако имеется и центральный компонент, который содержит метаинформацию и в случаях необходимости улучшения качества системы может корректировать поведение любого компонента структуры системы управления.

Реализация базовых нейрологических модулей

Базовыми нейрологическими модулями являются клетки и ядра, поскольку именно они позволяют формировать путем обучения функции, которые здесь считаются основными элементами при построении процессов управления. На этих базовых модулях строятся локальные сети, которые являются преобразователями образов, и распределенные сети, в которых формируются процессы, как последовательности преобразований образов.

Клетки и ядра, построенные в соответствии с приведенными здесь структурами, могут быть реализованы в интервально-логическом, нечетко-логическом или вероятностно-логическом базисах. Далее рассмотрим возможные реализации клеток и ядер в нечетко-логическом базисе, который представляется нам наиболее подходящим для построения процессов управления.

Базовые модули должны иметь обрабатывающую и обучающую части. Обрабатывающая часть модуля является носителем функции, которая конкретизируется настройкой параметров активатора (весов связей между элементами сети активатора). Настройка этих параметров производится настройщиком в соответствии с алгоритмом обработки примеров конкретной функции.

Обрабатывающая часть модуля, реализованная в нечетко-логическом базисе, имеет информационный гранулятор, построенный на принципе фазификации, активатор с нечетко-логическими вычислительными элементами и информационный дегранулятор, построенный на принципе дефазификации.

фикации (рис.3).

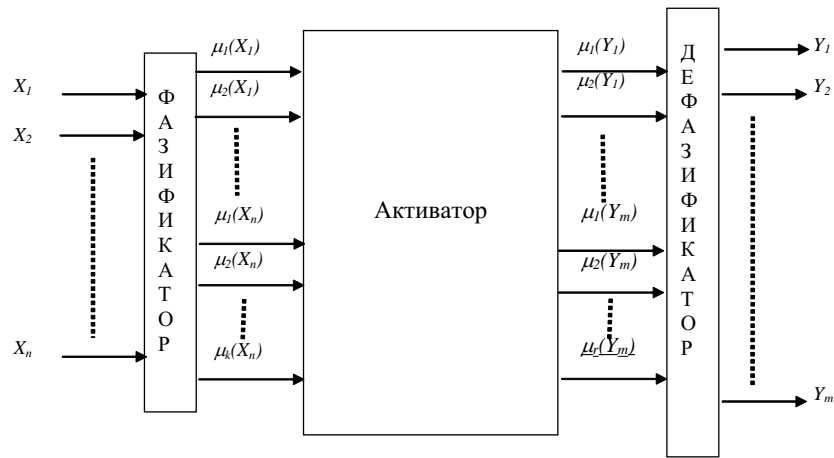


Рис. 3. Обработывающая часть нейробиологического модуля

Разработаны и исследованы три модели активатора: проекционная, сеточная и кластерная.

Проекционная модель

Проекционная модель является самой простой из разработанных моделей (рис.4).

Здесь $\mu_j(x_i)$ — значение j -й функции принадлежности i -го входного сигнала, $\mu_i(y)$ — значение i -й функции принадлежности выходного сигнала. Функция, реализуемая активатором, может быть представлена, как

$$\mu_m(y_r) = \bigcap_{i=1}^n \bigcup_{j=1}^{k_i} \mu_j(x_i) w_{l+m,p+j}. \quad (11)$$

Здесь введены следующие дополнительные обозначения: n — число входов, определяющее размерность входного вектора; w_{ij} — весовой коэффициент связи; k_i — число термов i -го входа; l — индекс, соответствующий первому терму r -го выхода; p — индекс, соответствующий первому терму i -го входа.

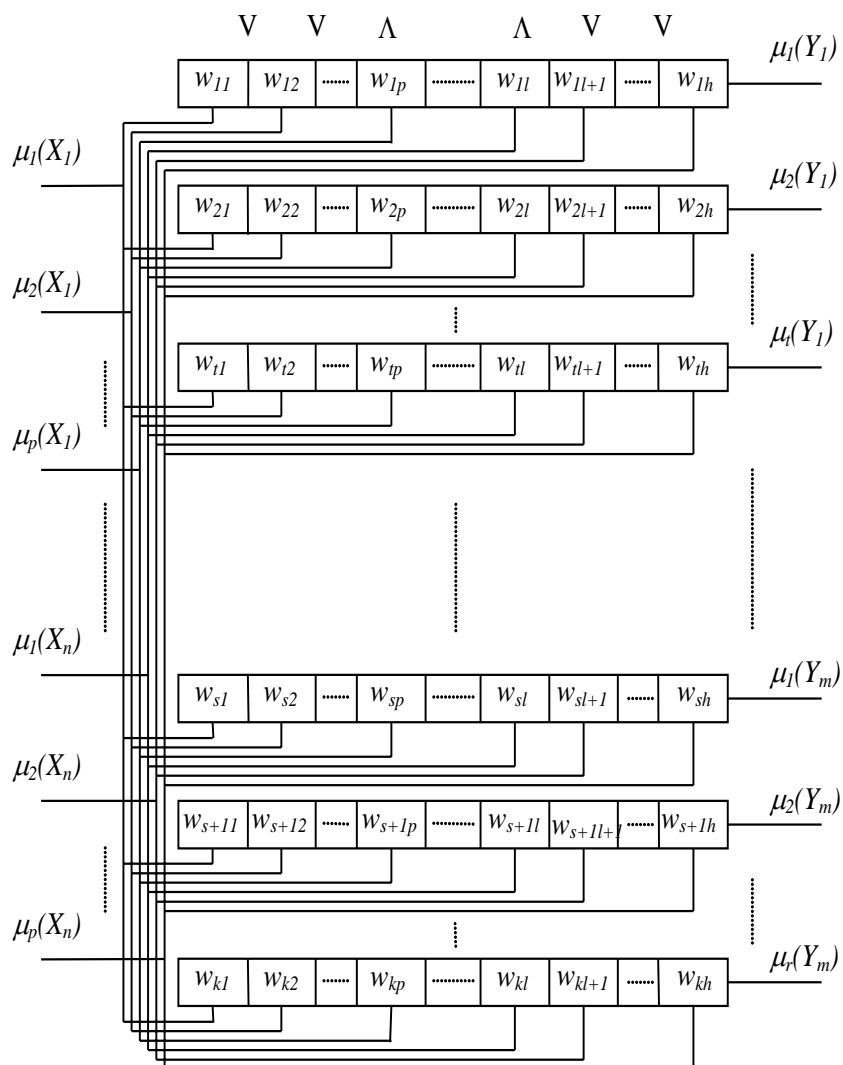


Рис. 4. Проекционная модель активатора

Наиболее эффективный метод обучения нейробиологического модуля с проекционной моделью активатора — *генетический алгоритм*. Процесс обучения представляет собой оптимизацию путем направленного случайного поиска весовых коэффициентов связей активаторных элементов с применением операторов кроссовера, мутации и селекции.

Преимуществами этой модели являются:

- низкие требования к используемой памяти весовых коэффициентов;
- простота структуры активатора;
- хорошая способность к обобщению при аппроксимации функций.

Эта модель, однако, имеет серьезные ограничения по применимости. Она не может быть обучена тем преобразованиям образов, для которых полная информация не может быть запомнена в проекциях. Кроме того, для обучения в этой модели невозможно использовать относительно простые градиентные алгоритмы.

Сеточная модель

Сеточная модель является универсальной в плане способности реализации практически любых преобразований входного вектора в выходной (рис.5).

Здесь $\mu_{ki}(Xi)$ является значением k_i функции принадлежности i -го входного сигнала, $\mu(y)$ — значение функции принадлежности выходного сигнала (показан случай векторного входа и скалярного выхода). Функция принадлежности, реализованная активатором, может быть представлена в следующей форме

$$\mu(y) = \bigcup_{k_1 \dots k_n}^{m_1 \dots m_n} \left(w_{k_1 \dots k_n} \bigcap_{i=1}^n \mu_{k_i}(X_i) \right). \quad (12)$$

Здесь введены следующие дополнительные обозначения: n — число входов, определяющее размерность входного вектора; $w_{i, \dots, j}$ — весовые коэффициенты связей активаторных элементов; m_i — число термов i -го входа.

Для обучения в такой модели может быть использован простой алгоритм вычисления весов связей. Этот алгоритм будет показан на примере нейробиологического модуля, который будет рассматриваться далее.

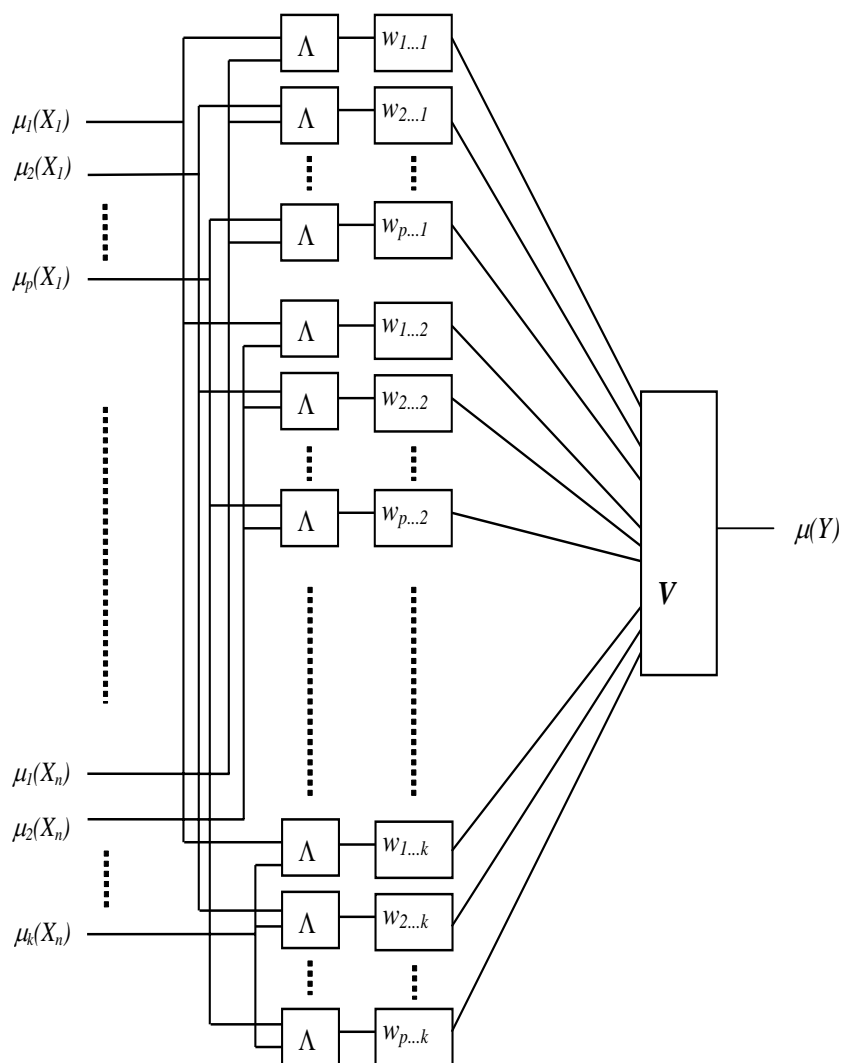


Рис. 5. Структура активатора для сеточной модели

Преимуществами модели являются:

- простота структуры и обучающего алгоритма;
- возможность обучения в реальном времени;
- способность аппроксимации практически любой непрерывной функции.

Модель имеет недостатки в плане избыточности внутренней структуры, высоких требований к размеру требуемой памяти и низкой способности к обобщения при аппроксимации функций.

Кластерная модель

Кластерная модель активатора может быть представлена структурой рис.6.

Эта модель не требует предварительного определения набора термов входных переменных, поскольку они формируются динамически при обработке примеров отображения функции. Более того, при использовании кластерной модели можно обойтись без дегранулятора и посылать сигнал, формируемый активатором непосредственно на выход модуля. В этом случае можно проиграть в точности аппроксимации, но использовать значительно меньшие потребляемые ресурсы (память и вычислительную мощность).

Здесь также $\mu_j(x_i)$ означает значение j -й функции принадлежности i -го входного сигнала. Функция, реализуемая активатором, может быть представлена в следующем виде

$$Y = \text{sign}(w_k) \bigcup_{i=1}^m \left| w_i \bigcap_{j=1}^n \mu_{ij}(X_j) \right|. \quad (13)$$

Здесь введены следующие дополнительные обозначения: n — число входов, определяющее размерность входного вектора; m — число примеров числовых отображений функции; w_i — весовые коэффициенты связей элементов активатора для i -го терма; k — число определяющих термов.

Настройка весовых коэффициентов и координат границ и центров определяющих термов реализуется путем обучения с использованием специального алгоритма.

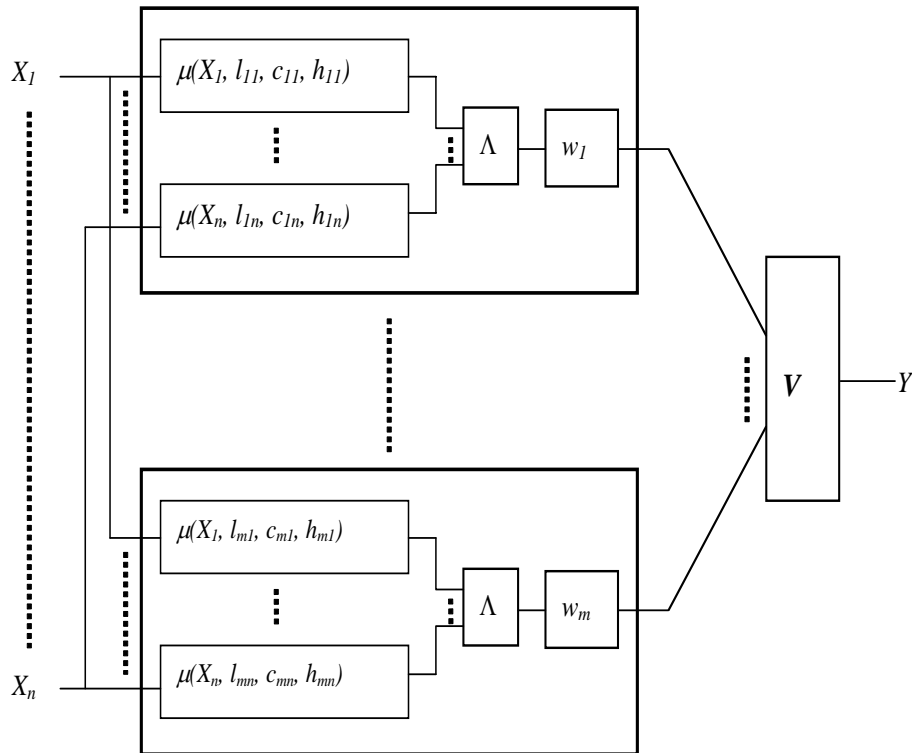


Рис. 6. Структура активатора для кластерной модели

Преимущества модели являются:

- простота структуры и обучающего алгоритма;
- способность аппроксимации практически любой непрерывной функции;
- хорошая способность к обобщению при аппроксимации; низкие требования к расходованию памяти.

В качестве недостатка можно отметить возможные сложности при аппаратной реализации нейрологических модулей с использованием этой модели.

Исследования показали, что кластерная модель является наиболее пригодной для построения нейробиологических модулей. Такая модель комбинирует преимущества проекционной и сеточной модели и является универсальным адаптивным аппроксиматором с низкими требованиями к расходуемой памяти.

В табл. 1 приведены *оценки представленных моделей* по критериям:

- вычислительной сложности при обработке информации;
- требованиям к размеру памяти;
- точности отображения функций.

Предполагается, что для управляющих применений вычислительная сложность является наиболее важным критерием, за которым следуют требования к памяти и точность отображения функции.

Таблица 1. Сравнительная оценка свойств проекционной, сеточной и кластерной моделей

	Проекционная модель	Сеточная модель	Кластерная модель
Сложность	$2p \cdot n \cdot \Delta t_1 + 3p \cdot n \cdot \Delta t_2$	$p \cdot 2^n \cdot n \cdot \Delta t_1 + p \cdot 2^n \cdot (n+1) \cdot \Delta t_2$	$\sqrt{e} \cdot n \cdot \Delta t_1 + \sqrt{e} \cdot n \cdot \Delta t_2$
Объем памяти	$p \cdot \sum_{k=1}^n m_k$	$p \cdot \prod_{k=1}^n m_k$	$3 \cdot e \cdot n$
Точность	Невозможно отображение сложных функций	Возможно отображение сложных функций в режиме интерполяции	Возможно отображение сложных функций с простой интерполяцией

В таблице введены следующие обозначения: n — число входов; m_k — число термов k -го входа; p — число термов выходного сигнала; e — число примеров числового отображения функции; Δt_1 — время оценки функции принадлежности; Δt_2 — максимальное/минимальное время оценки.

Моделирование на PC Pentium с тактовой частотой 1 ГГц выявило следующие допустимые диапазоны для введенных факторов: n : $4 \div 12$; m_k : $5 \div 7$; p : $8 \div 12$; e : $200 \div 2000$; $\Delta t_1 = 10 \cdot \Delta t_2$.

Когнитивные нейрологические системы управления и агенты

Понятие когнитивности (cognition) давно используется в когнитивной психологии и связывает концепции познания и знаний [14]. Когнитивная психология детально изучает информационные процессы мозга человека. Предполагается, что мозг воспринимает информацию, т. е. обрабатывает ее сенсорными системами и формирует структурированную систему знаний. Знания сначала фиксируются в кратковременной памяти, а затем накапливаются и хранятся в долговременной памяти в кодированном виде. Формирование и накопление знаний связано с концепцией познания. Концепция мышления, т. е. ментальной обработки информации, определяет процесс использования знаний решения различных задач. Результаты исследования процессов обработки информации, связанных с познанием и мышлением, позволили построить *когнитивную теорию*, описывающую работу мозга на основе информационного подхода.

Специалисты по интеллектуальным системам, использующие когнитивную теорию, стали трактовать познание с технической точки зрения, как способность системы автоматически извлекать знания и накапливать их в памяти в процессе обучения. Мышление трактуется, как использование знаний и выводов на них для формирования рационального поведения системы. При этом используются аналогии ментальных операций, связанные с выводом путем рассуждений или ассоциативным выводом. В рамках когнитивной теории были выделены и частично изучены основные *когнитивные функции и процессы*, определяющие познание и мышление. Эти результаты оказались полезными при разработке *когнитивных систем*, которые должны обеспечивать решение сложных задач обработки информации и управления на основе моделирования когнитивных функций и процессов, протекающих в мозге человека при решении подобных задач. Реальные успехи в создании систем, способных реализовать даже уже известные исследователям когнитивные функции и процессы мозга, пока небольшие. Трудности определяются сложностью и недостаточной изученностью этих функций и процессов, а также ограниченными возможностями используемых вычислительных структур, таких как логические или нейронные сетевые вычислители.

Сложное поведение объектов можно обеспечить, создавая *когнитивные системы управления* на основе формализованных методов обучения и ментального решения задач [1, 2]. Эти методы позволяют строить обу-



Рис. 7. Управляющие структуры когнитивной системы

чаемые модули для работы с когнитивными функциями — *когнитивные модули* — и сети таких модулей для организации когнитивных процессов — *когнитивные структуры*. Когнитивные структуры позволяют накапливать в процессе обучения необходимые знания и выбирать ментальным образом нужные действия при организации целесообразного индивидуального и коллективного поведения (рис.7). Оказалось полезным выделить в отдельную группу функции управления силовыми приводами (актуаторами) динамических объектов. Они, как и когнитивные функции, тоже могут быть реализованы на обучаемых компонентах. Можно выделить также процессы управления системой актуаторов объекта, которые здесь условно названы актуаторными процессами. По аналогии с когнитивными структурами, введем актуаторные структуры (АС), способные обучаться и реализовать процессы в системе актуаторов объекта. Эти процессы, как правило, связаны

с обработкой заданий на координированное управление исполнительными устройствами объекта.

Когнитивные системы управления, построенные в соответствии с изложенными принципами, могут в общем случае включать выделенные ветви: восприятия информации о среде, генерации поведения (цепочек управляемых действий) с учетом текущей ситуации и координированного исполнения действий (рис. 7). Координация выделенных ветвей требует реализации когнитивных процессов их взаимодействия с помощью специальных когнитивных структур. Таким образом, в когнитивных системах при управлении объектами реализуются цепочки преобразований следующего вида:

$$\{E \rightarrow S \rightarrow CSP \rightarrow CSB \rightarrow B \rightarrow CSB \rightarrow CSC \rightarrow AS \rightarrow A \rightarrow R \rightarrow E\},$$

где E — среда, S — сенсоры, CSP — КС «Восприятие», CSB — КС «Восприятие-Поведение», B — выбранные поведенческие компоненты, CSB — КС «Восприятие-Поведение-Управление», CSC — КС «Поведение-Управление», AS — АС координации управления, A — актуаторы, R — робот, E — среда.

Когнитивные и актуаторные структуры предлагается строить как сети нейрологических модулей, соединенных прямыми и обратными связями. В когнитивных системах управления когнитивные структуры должны воспринимать цели, ограничения и вырабатывать сигналы селекции поведения в зависимости от текущей ситуации (состояния объекта управления и внешнего мира). На выходах каждой когнитивной структуры формируются сигналы активизации выбранного поведения. Актуаторные структуры должны воспринимать задания на перемещение объектов управления или взаимодействия с другими объектами среды и вырабатывать сигналы управления отдельными актуаторами. Компонентами структур являются модули типа клеток, ядер и локальных сетей с настраиваемыми путем обучения параметрами. Связи между компонентами отображают передачи выработанных ими сигналов другим компонентам или обратных сигналов от них.

Разработки в области когнитивной теории мозга и недавние исследования по нейрофизиологии позволили сформировать новую *гипотезу о строении и функционировании нервной системы*, связывающую эти области знаний о ней. Так, нейрофизиологи частично прояснили локализацию некоторых когнитивных процессов в нейронных структурах мозга [15]. Показано, что в зависимости от сложности когнитивных функций и процессов в мозгу используются локальные, проекционные и перекрывающиеся нейронные сети. Кроме того, показано превалирующее значение замкнутых

подсистем внутри нервной системы человека, которые реализуют поведенческие процессы. Эти подсистемы представляют собой цепочки нейронных модулей (локальных сетей), распределенные по ряду областей нервной системы, включая мозг и сенсомоторные области. Такие цепочки постоянно активны, но только, если их активность высока по отношению к другим, они выполняют свои поведенческие функции реально. Они пересекаются и взаимодействуют между собой в перекрывающихся областях мозга. Объединения таких цепочек можно считать *агентами нервной системы*. Внутри объединения можно выделить когнитивные и актуаторные структуры, формирующие и исполняющие целенаправленные согласованные поведения агентов нервной системы. Агенты могут конкурировать между собой или кооперироваться для организации сложного поведения. Коллективная работа сообщества агентов, подчиненная некоторым «социальным» законам нервной системы, приводит к ее оптимальному функционированию. Перекрывающиеся области мозга выполняют роль социального регулятора, коммутирующего связи агентов и координирующего их действия. По существу, это мета-агент, который видит больше других и улучшает работу сообщества агентов системы. С помощью когнитивных и актуаторных структур агент извлекает информацию о среде от своих или используемых совместно с другими сенсоров, активизирует нужные обрабатывающие функции в своих наборах микроколонок нейронов коры мозга (нейронных модулях) и управляет своими или совместными эффекторами для воздействия на среду. Агенты самоорганизуются за счет генетической информации и могут самосовершенствоваться в процессе эволюции.

Такая гипотеза послужила основой для разработки методологии создания *многоагентных когнитивных систем управления* [2]. Структура многоагентной когнитивной системы включает три рассмотренные ранее ветви обработки информации, связанные пересекающимися когнитивными структурами, которые организуют взаимодействия между выделенными виртуальными агентами (рис. 8).

Нейрологическими компонентами такой системы являются ядра (N — Nucleuses), составленные из параллельно включенных клеток. Они могут обучаться простым когнитивным или аффективным функциям. Из ядер формируются локальные сети (LN — Local Networks) и распределенные кустовые сети проекционного уровня (PN — Projective Networks), работающие с образами. Объединение когнитивно-аффективных процессов, реализованных на соответствующих компонентах, образуют внутренние *когнитивные агенты* (ca — cognitive agent), которые, за счет настройки функций

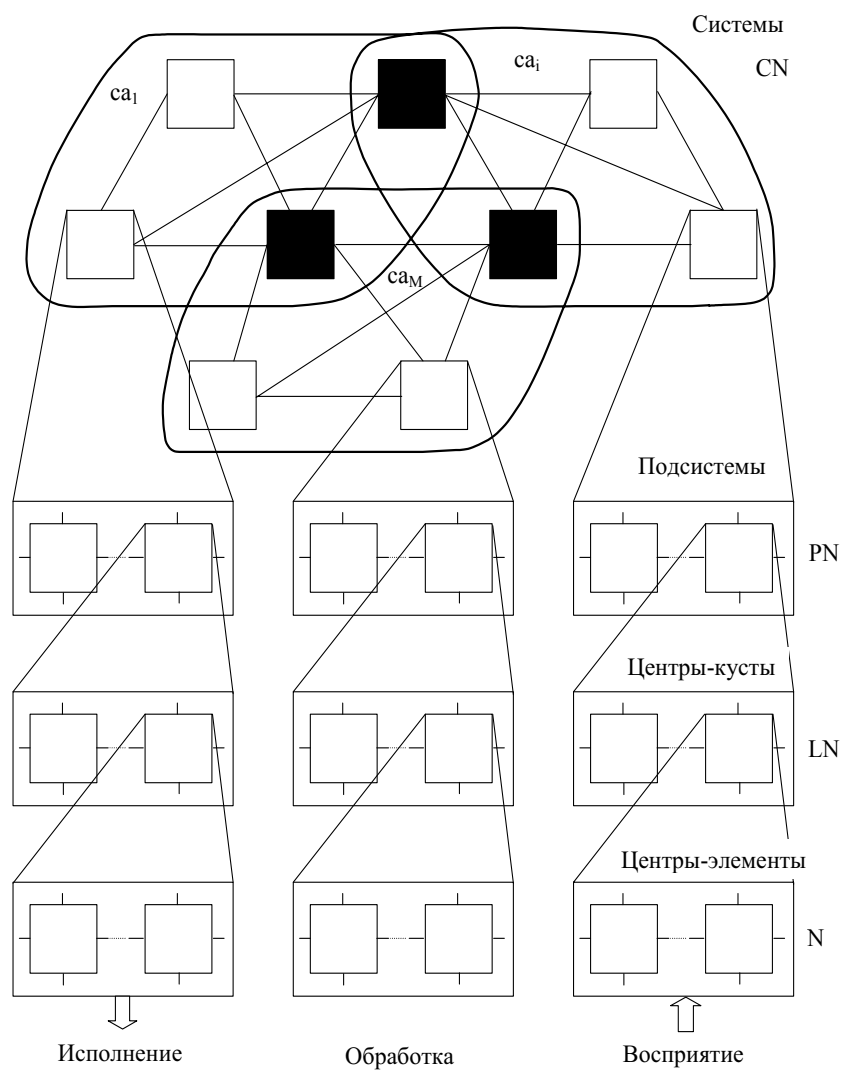


Рис. 8. Структура многоагентной когнитивной системы

восприятия, обработки и исполнения путем обучения, обеспечивают определенное поведение робота. Внутренние агенты взаимодействуют между собой через общие обучаемые компоненты уровня перекрывающихся сетей (CN — Cross Networks). В общем случае такая многоагентная когнитивная система управления может быть организована с несколькими уровнями (центры-элементы и кусты, подсистемы и системы).

Когнитивные агенты составляют основу когнитивной системы управления, построенной по многоагентной технологии [3]. Предшествующие разработки интеллектуальных управляющих агентов, таких как ATLANTIS [16], не имели компоненты обучения в процессе работы. Первые попытки разработки интеллектуальных обучаемых или когнитивных агентов и систем привели к необходимости разработки специальных *когнитивных архитектур* и математического аппарата для их описания и конструирования. Когнитивные архитектуры представляют собой модели человеческих рассуждений. Примерами ранних когнитивных архитектур являются системы ACT [17] и SOAR [18]. Они основаны на продукционных системах. В них рабочая память моделирует кратковременную память человека, а память продукции является частью долговременной памяти. Обе системы имеют сложные механизмы разрешения конфликтов и сохранения результатов в форме новых продукции, которые могут быть использованы, чтобы избежать подобных выводов в будущих ситуациях.

Более поздние разработки по когнитивным архитектурам базировались на абстрактных подходах с использованием модальной, динамической и темпоральной логик. В последних вариантах когнитивных архитектур предлагалось включение нескольких уровней обработки информации. Верхний уровень знаний, соответствующий модальному подходу, может быть неточным. В то же время, более низкий символичный уровень, соответствующий подходам с символическим представлением знаний, должен быть более точным, поскольку здесь используются логические процедуры вывода. Наиболее интересной когнитивной архитектурой является BDI (Belief-Desire-Intention)-архитектура [19]. Она использует такие когнитивные концепции, как Убеждение, Желание и Намерение, а также специальную BDI-логику для манипулирования ими.

Перспективными являются когнитивные агенты комбинированной архитектуры с многослойной структурой и многоуровневой организацией поведения. При использовании нейробиологических средств такие агенты становятся обучаемыми и даже самообучаемыми. Кроме того, поведение агентов может быть очень сложным, подобным человеческому. Примерами так

построенных агентов являются: *агент-футболист*, разработанный для игровой среды футбола роботов (RoboCup), а также *агент-пилот*, разработанный для симуляции воздушных операций в среде с противодействием [20]. Первый агент имеет реактивную архитектуру и трехуровневую организацию поведения (навыки, индивидуальное поведение и координация в команде). Агент-пилот имеет комбинированную архитектуру: нижний реактивный уровень реализует с помощью многослойной структуры простые исполняемые поведения, средний BDI-уровень генерирует текущее поведение в соответствии с индивидуальным намерением агента, а верхний BDI-уровень обеспечивает координацию с целью согласования намерений взаимодействующих агентов и выработки их общего намерения, которое используется для коррекции индивидуального намерения агента.

На основе исследований упомянутых программных агентов разработан ***общий метод организации многоуровневого поведения когнитивных агентов***, управляющих динамическими объектами, с использованием когнитивных структур, реализованных на нейробиологических средствах. Предполагается, что в общем случае когнитивный агент должен иметь, как минимум, трехуровневое поведение: нижний уровень реализует примитивные поведения динамического объекта управления, средний уровень обеспечивает сложное индивидуальное поведение объекта, верхний уровень отвечает за коллективное поведение объекта при взаимодействии его с другими объектами при выполнении общих заданий.

На верхнем уровне агента используется когнитивная структура (рис.9), построенная как сеть обучаемых модулей, которые формируют набор когнитивных функций (CF), определяющих когнитивные процессы организации взаимодействий роботов в группе. Эта когнитивная структура включает три уровня: групповых целей, обмена сообщений и согласования действий.

Когнитивные функции, обрабатываемые в первом слое структуры, имеют первыми аргументами общие цели (G) группы агентов, которые можно рассматривать, как коллективные «желания». Вторыми аргументами этих функций являются состояния мира объекта (WS) и его внутренние состояния (IS), которые формируются в результате обработки текущей сенсорной информации в модуле модели мира (WM) и информации от внутренних датчиков в модуле внутренней модели (IM). Эта информация может рассматриваться, как «убеждения» агента, которые могут изменяться во времени. В результате обработки аргументов когнитивные функции возвращают сигналы селекции необходимых поведений агента при работе в группе.

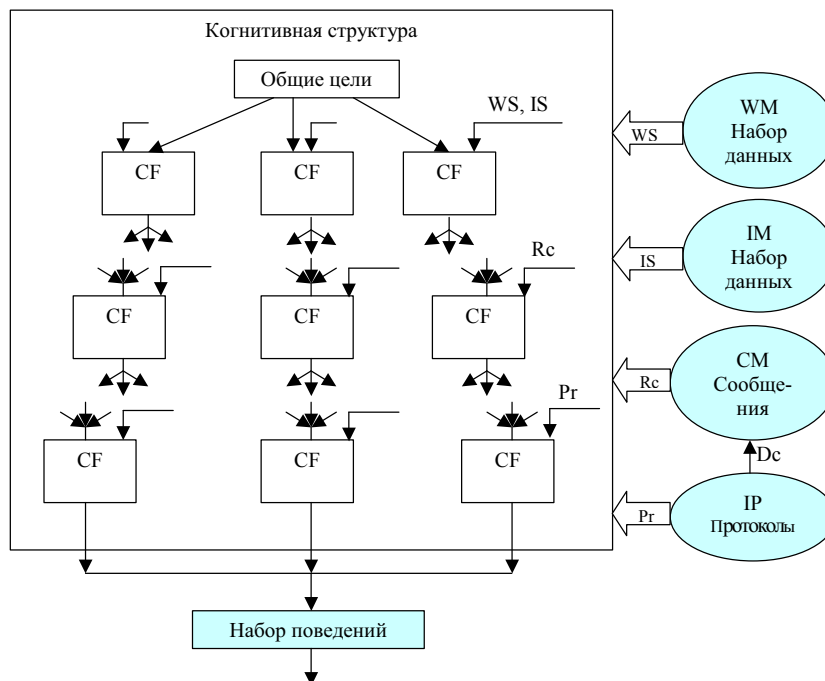


Рис. 9. Когнитивная структура верхнего уровня агента

Для этого они должны быть настроены (обучены) классификации текущих ситуаций среды и принятию решений о текущих целях и соответствующих действиях объекта, управляемого агентом. Здесь возможно обучение с учителем, фиксирующим примеры правильной классификации ситуаций и нужных действий, или же самообучение с использованием подкреплений, автоматически формируемых по критерию достижимости желаемых целей агента.

Когнитивные функции, реализуемые во втором слое структуры, в качестве первых аргументов воспринимают текущие ситуации и цели, полученные первым слоем. Вторыми аргументами каждой из когнитивных функций являются признаки (R_c), формируемые путем обработки полученных сообщений (приходящих от других агентов, в том числе и в ответ на запросы данного агента) модулем коммуникаций агента (CM). Возвраща-

щаемыми значениями этих когнитивных функций являются текущие цели и действия, откорректированные с учетом принятых сообщений от других агентов. Когнитивные функции второго слоя, как правило, формируются путем обучения с учителем, который отбирает, как эксперт, правильные реакции на сообщения.

Когнитивные функции, сформированные в третьем слое структуры, в качестве первых аргументов имеют возвращаемые значения когнитивных функций первого и второго слоев. Вторыми аргументами являются сигналы синхронизации протокола взаимодействий (Pt), которые вырабатываются модулем протокола взаимодействий (IP). Возвращаемыми значениями этих функций являются текущие цели и действия по коммуникациям с другими агентами (формирование сообщений в соответствии с протоколом согласований намерений агента с другими агентами). Когнитивные функции этого слоя формируются путем обучения по примерам правильного взаимодействия в соответствии с принятым протоколом.

В результате последовательной обработки когнитивных функций в когнитивной структуре верхнего уровня агента реализуются когнитивные процессы формирования текущих целей и соответствующих планов действий, которые можно рассматривать, как «намерения» агента, согласованные с намерениями других агентов. Эти намерения инициируют соответствующее коллективное поведение агента из набора взаимодействующих поведений, которые умеет выполнять агент. Такие поведения можно рассматривать, как «коллективные умения» агента. Согласованные намерения агента передаются на средний уровень когнитивного агента (он не показан на рисунке).

Средний уровень агента имеет свою когнитивную структуру, обрабатывающую информацию о состоянии мира объекта и его внутреннем состоянии (WS, IS) и формирующую текущие цели и соответствующие действия агента (индивидуальные намерения), руководствуясь либо согласованными намерениями, сформированными на верхнем уровне, либо общими целями агента без согласований с другими агентами. Индивидуальные намерения инициируют соответствующее индивидуальное поведение агента из набора поведений, которые умеет выполнять агент (индивидуальные умения).

Нижний уровень структуры агента может иметь свою когнитивную структуру, которая обрабатывает внутренние состояния объекта (IS) и, в соответствии с намерениями, выработанными на среднем уровне, формирует сигналы выбора навыков, которым обучен агент. Эти навыки выбираются из набора навыков (примитивных умений), которые предназначены для

реализации конкретных действий управляемого объекта, инициированных в результате работы всех уровней структуры агента.

Когнитивные структуры всех уровней агентов, которые также являются когнитивными системами управления, могут быть реализованы на описанных здесь нейробиологических средствах. Компонентами таких структур являются модули-ядра (рис.2), в которых путем обучения формируются когнитивные функции. Локальные сети из таких модулей образуют когнитивные структуры, которые реализуют определенные когнитивные процессы выбора поведения по текущей ситуации. Эти структуры настраиваются также путем обучения.

В качестве примера рассмотрим *структуру модуля-ядра*, построенную на основе сеточной модели и реализованную в программном агенте-футболисте (рис.10). Модуль реализует нечетко-нейронное (In-Out) отображение входного n -мерного вектора X , в выходной m -мерный вектор Y и его просмотр (Look). Обработывающая часть (Process Unit) включает гранулятор (fuz), активатор, состоящий из блока ассоциативного вывода, реализующий процедуру взвешенной агрегации и объединения (Inference), и памяти весов (Memory), и дегранулятор (dfuz). Обучающая часть (Learning Unit) включает два гранулятора (fuz), блок вычисления текущих весов (Current Weight), реализующий процедуру коррекции весов (w-update), и регистры (Reg) для образов (Pattern). Используются следующие внутренние процедуры:

- $\text{fuz}(x_i, \mu^q(x_i))$ — фазификация, вычисляющая степень принадлежности переменной x_i к q -й нечеткой грануле этой переменной;
- $\text{inference}(ln, w_i, \mu_{xi}^q; i = 1, \dots, I; q = 1, \dots, K)$ — взвешенная агрегация, вычисляющая каждую степень принадлежности переменной y_j к q -й нечеткой грануле этой переменной путем агрегирования степеней принадлежности входных переменных x_i , с дальнейшим умножением результата на вес w_i и объединением; I, K — количество агрегируемых степеней принадлежности и нечетких гранул соответственно; ln — номер слоя сети, в котором выполняется агрегация;
- $\text{dfuz}(\mu_{yj}^q, y_j; q = 1, \dots, K; j = 1, \dots, M)$ — дефазификация, вычисляющая значение выходного параметра y_j , исходя из степеней принадлежности значения этого параметра к нечетким гранулам шкалы параметра;
- $\text{w-update}(w_p, \mu_y^*, \mu_y, p)$ — коррекция веса при обучении; w_p — текущий вес; μ_y^q, μ_y — степени принадлежности правильного и текущего

значений выходного параметра y к нечеткой грануле соответственно (правильное значение параметра берется из примера для обучения); p — номер примера; в результате операции находится веса при обработке $p + 1$ примера.

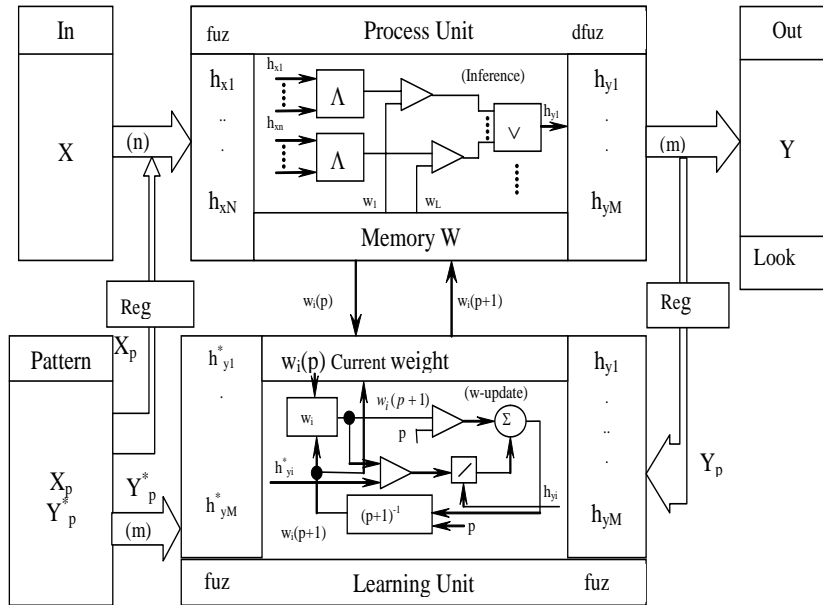


Рис. 10. Структура нейрологического модуля

Рассмотрим пример конкретизации процедур **fuz**, **inference**, **dfuz**, **w-update** для случая двух входных переменных x_1, x_2 и одной выходной переменной y . Пусть каждая переменная декомпозируется на три нечетких гранулы и функции принадлежности значений переменных к гранулам имеют треугольный вид. Тогда указанные процедуры примут форму ($q = 1, 2, 3$

— номер гранулы на шкале):

$$\begin{aligned} \text{fuz} : \mu_x^q &= ((x - l_q)c_{ql}^{-1} \Leftarrow l_q < x < m_q) \\ &\vee ((r_q - x)c_{qr}^{-1} \Leftarrow m_q < x < r_q) \vee \end{aligned} \quad (14)$$

$$\begin{aligned} (0 \Leftarrow \text{else}); c_{ql} &= m_q - l_q; c_{qr} = r_q - m_q, \\ \text{inference} : \mu_y^q &= \bigvee_{i=1}^9 (w_i^q \wedge_{q,r=1}^{q,r=3} (\mu_{x1}^q, \mu_{x2}^r)), \end{aligned} \quad (15)$$

$$\begin{aligned} \text{dfuz} : y^q &= 0.5(y_n^q + y_r^q) = 0.5(\mu_y^q/c_{qn} + \mu_y^q/c_{qr}); \\ y &= \Sigma_{q=1}^K \mu_y^q y^q / \Sigma_{q=1}^K \mu_y^q, \end{aligned} \quad (16)$$

$$\begin{aligned} \text{w-update} : w_{p+1} &= w_p(p/p + 1) + w_p(\mu_y^*/\mu_y)/(p + 1); \\ p &= 1, \dots, n_p. \end{aligned} \quad (17)$$

Здесь введены дополнительные обозначения: l_q , m_q , r_q — координаты левого основания, вершины и правого основания треугольной функции принадлежности; \wedge — нечеткая операция пересечения множеств (\min), примененная к парным комбинациям степеней принадлежности значений входных переменных к q -й грануле; \vee — нечеткая операция объединения множеств (\max), примененная к 9-ти парным результатам парных операций пересечения с учетом их весов.

Рассмотренный подход к построению когнитивных систем управления и агентов на базе нейробиологических средств в настоящее время используется при разработке систем управления такими сложными динамическими объектами, как интеллектуальные роботы. Создание эффективных систем управления для таких роботов сейчас является одним из самых актуальных направлений в связи с бурным развитием работ в области человекоподобных роботов [5]. Прогнозы показывают, что в ближайшей перспективе рынок таких роботов может конкурировать даже с рынком автомобилей.

Интеллектуальные роботы и системы управления

По мере развития искусственного интеллекта возникло естественное стремление к переходу от создания изолированных эвристических и интеллектуальных программ к разработке систем, которые могли бы решать интеллектуальные задачи при активном взаимодействии со средой. Это привело к созданию *интеллектуальных роботов*, в которых сведены в единый комплекс интеллектуальные средства принятия решений, а также сенсорные и моторные системы.

В начальный период проводились работы по созданию, так называемых, *интегральных роботов* (систем «глаз-рука», «глаз-рука-ухо», «глаз-тележка»). Понятие «интегральный робот» впервые было введено *Н. Нильсоном* применительно к устройству «Шейки», разработанному и исследовавшемуся Стенфордском исследовательском институте в период 1966–1972 годов [4]. Устройство представляло собой тележку, имеющую сенсоры типа «глаза» и тактильных щупов и способную перемещаться и передвигать предметы. В дальнейшем в лабораториях США, Англии, Японии и других стран было создано и исследовано множество вариантов интегральных роботов. В России тоже разрабатывались и исследовались подобные роботы. Один из них, ЛПИ-3, был оснащен системой технического зрения, тактильными сенсорами на схватах двух рук, голосовой командной системой [7]. Исследования в области интегральных роботов серьезно продвинули теорию интеллектуального управления, а также дали много эффективных решений для разработки наземных, подводных и космических робототехнических комплексов.

В настоящее время к интеллектуальным роботам могут быть отнесены многочисленные варианты реальных мобильных и виртуальных программных роботов с интеллектуальным управлением. *Мобильные роботы*, предназначенные для работы в одиночку (в экстремальных условиях, при обслуживании больных и детей, в охранных системах и пр.), часто выпускаются серийно и имеют относительно простые системы интеллектуального управления. Мобильные роботы, способные работать в группе (разведывательные, обеспечивающие связь, боевые, охранные, игровые и пр.), еще только осваиваются и имеют сложные системы интеллектуального управления. *Программные роботы* (softrobots) стали разрабатываться недавно для применений в качестве информационных или персональных помощников, поисковых интернет-агентов, виртуальных игровых агентов и пр. Наиболее сложными являются игровые роботы-агенты, например, для соревнований на Кубок роботов (RoboCup).

Многочисленные попытки разработки мобильных *антропоморфных роботов* (андроидов), которые были названы так за внешнее сходство с человеком, позволили разработать основные положения теории ходьбы механизмов на двух ногах [6]. Однако успехи в практической реализации таких роботов были очень незначительными. В последующие годы происходило постоянное совершенствование, как конструкций двуногих шагающих роботов, так и их систем управления. Наконец, в начале XXI века, благодаря последним технологическим достижениям мехатроники, микроэлектрони-

ки, разработкам в области интеллектуального управления роботами, были созданы первые *гуманоидные роботы*. Именно так стали теперь называть автономные шагающие человекоподобные роботы нового поколения [5, 6]. Создание этих роботов ознаменовало серьезный успех искусственного интеллекта и робототехники, знаменующий начало эры полноценных помощников человека, похожих на него не только по форме, но и по поведению.

Система управления первого интеллектуального робота «Шейки» была многоуровневой и осуществляла интегральную обработку сенсорной информации, формирование на ее основе модели внешней среды, планирование последовательных действий для решения задач в данной среде и их выполнение с дополнительным планированием на более низких уровнях. То, что робот мог взаимодействовать непосредственно с реальной средой, имело принципиальное значение для синтеза его системы управления. Это позволило моделировать поведение биологических объектов путем системного рассмотрения реализующих это поведение процессов в рамках замкнутого поведенческого акта — от формирования целей к их достижению за счет планирования и выполнения последовательности преобразований проблемной среды. Реализация такой системы потребовала использования интеллектуальных методов, поэтому системы такого рода называли также называть *интеллектуальными системами управления*.

Традиционно интеллектуальные системы управления мобильными роботами строились иерархическими, например, с 5-ю уровнями: стратегическим, тактическим, адаптивным, программным и исполнительным. При этом использовалась гипотеза о пирамидном строении нервной системы человека [7]. На верхних уровнях принимается и обрабатывается задание от оператора, а на нижних — формируется множество управлений приводами степеней подвижности робота. При этом верхний стратегический уровень анализирует задание и планирует деятельность робота для его выполнения. Тактический уровень осуществляет управление реализацией плана действий в соответствии с последовательностью операций, предусмотренных планом. Адаптивный уровень осуществляет коррекцию действий при изменениях условий внешней среды. Все перечисленные верхние уровни управления используют сенсорную информацию о среде. Нижний уровень программного управления реализует отдельные элементарные операции с помощью заранее подготовленного набора программ для их выполнения. Самый нижний уровень является исполнительным и непосредственно управляет приводами робота. Нижние уровни управления используют ин-

формацию от внутренних датчиков робота о положении степеней подвижности и состояниях его компонентов.

Такие системы, реализованные с использованием символистских методов, основанных на обработке символьных знаний, имели ограниченные возможности при решении задач управления сложным поведением и адаптации к среде. В современных системах управления интеллектуальными роботами используются, как символистские, так и коннективистские, основанные на нейронных сетях, методы решения задач управления. Первые позволяют планировать действия роботов, а вторые — строить обучаемые устройства распознавания образов, принятия решений и исполнения элементарных действий робота.

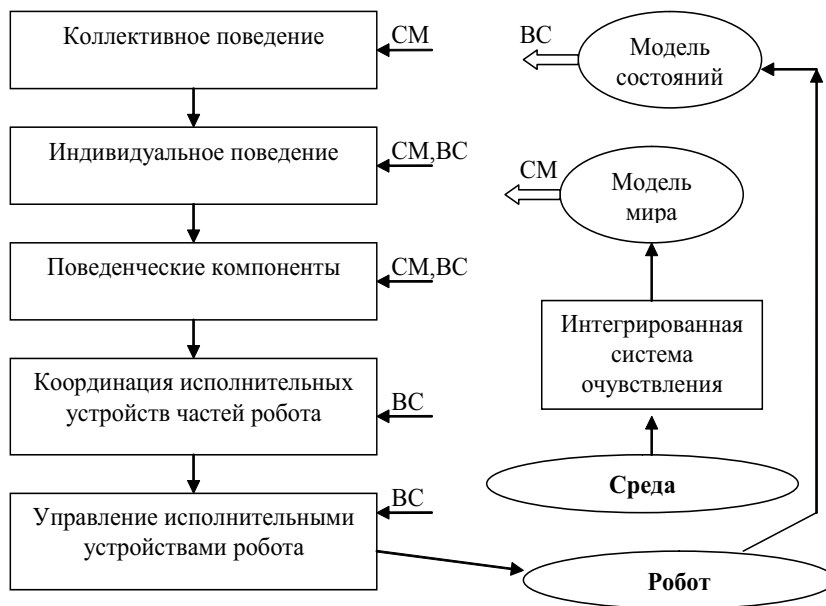


Рис. 11. Структура системы управления интеллектуальным роботом

Современные интеллектуальные мобильные роботы, способные работать совместно с другими роботами или человеком, имеют наиболее сложную многоуровневую систему управления (рис.11). Она должна включать Интегрированную систему очувствления, объединяющую системы виде-

ния, акустики, тактильного и силомоментного осязания, внутреннего состояния робота и др., и поставляющую информацию в модули Модели состояния и Модели мира робота для использования информации о Состоянии Мира (СМ) и Внутреннем Состоянии (ВС) всеми уровнями управления. Нижний уровень должен обеспечивать непосредственное управление исполнительными устройствами робота (приводами степеней подвижности). Второй уровень системы должен осуществлять координированное управление частями робота (устройствами перемещения, манипуляторами, сенсорными устройствами и пр.) при выполнении заданных движений. На третьем и четвертом уровнях необходимо выбрать и реализовать компоненты индивидуального поведения робота (восприятие объектов сцены, навигацию в среде, работы с объектами, отработка команд и программ) с использованием заготовленного набора поведенческих компонент. Для организации групповой работы роботов и целенаправленного безопасного функционирования в среде людей в системе управления должен быть уровень коллективного поведения, обеспечивающий выбор и реализацию функций, необходимых для координации поведений робота при работе в среде других роботов или людей.

Сложность поведения интеллектуальных роботов приводит к необходимости поиска нетрадиционных путей построения их систем управления. Один из таких путей состоит в использовании *принципов организации нервной системы человека*, раскрытых современной психологией и нейрофизиологией. Основопологающим принципом является *клеточное* строение нервной системы. Техническая реализация этого принципа может быть сведена к аппаратному или программному вариантам многоуровневых сетевых систем управления на однородных клеточных вычислителях (на формальных или неформальных нейронах или других обучаемых компонентах). Такие системы можно отнести к чисто коннективистским системам, поскольку в них не используются средства, основанные на обработке символьных знаний. По существу, такой подход связан с созданием так называемого компьютерного мозга робота.

Современные коннективистские системы управления интеллектуальных роботов, реализованные на базе компьютерного мозга, пока по возможностям значительно уступает человеческому мозгу. Так, японская корпорация Фудзицу еще в 1997 году объявила о создании нейрокомпьютерного мозга робота, эквивалентного по интеллекту системе из 100 биологических нейронов. Такой мозг успешно решает задачи распознавания образов и синергетического управления приводами робота, но его мощности недо-

статочно для организации сложного человекоподобного поведения робота при взаимодействии с людьми или другими роботами.

Значительное усиление интеллекта роботов может дать использование при создании систем управления роботов рассмотренных здесь *когнитивных принципов и нейрологических компонент*. Предполагается, что именно на базе нейрологических средств могут быть созданы когнитивные системы и агенты, которые позволят обеспечить сложное, близкое к человеческому, поведение современных интеллектуальных роботов гуманоидного класса.

Появление первых интеллектуальных роботов гуманоидного класса вызвало широкий интерес во всем мире. О создании гуманоидных роботов (ГР) объявили сразу несколько ведущих японских фирм и лабораторий: Honda — P3, Asimo; Sony — SDR-3 и 4; Fujitsu — HOAP-1; JSC Laboratory — H-7. Эти результаты являются следствием большого интереса и мощной поддержки таких работ со стороны правительства и ведущих фирм Японии. Так, фирма Honda финансирует и ведет работы по ГР, начиная с 1983 года. Роботы P3 и Asimo представляют 3-е поколение ГР фирмы. До них лаборатории фирмы разработали более 10-ти прототипов, начиная от двуногих платформ и кончая полностью автономными человекообразными роботами нормального и среднего роста (P3 имеет рост 180 см., Asimo — 120 см.). Фирма Sony разработала пока лучший в мире вариант малого ГР (SDR-4 имеет рост 53 см.). Этот робот — результат многолетних работ с Институтом Гуманоидных роботов (HRI) университета Waseda с привлечением ведущих ученых и робототехников мира. Робот SDR-4, который предполагается выпускать серийно, может делать практически полный набор человеческих движений и имеет достаточно сложное поведение, позволяющее ему общаться с людьми и выполнять простые работы. Система осязания и управления позволяет этому роботу распознавать лица и голоса людей, выполнять простые задания в известной ему среде, а также обмениваться сообщениями с людьми или другими ГР.

Первыми применениями гуманоидных роботов будут: уход за больными, выполнение домашних работ, подвижные игры, обслуживание в общественных местах. В дальнейшем, не исключено их применение для ведения боевых операций, борьбы с террористами, охраны и т.д. Созданная в 1997 году федерация RoboCup, которая организует научные симпозиумы и соревнования на Кубок роботов, провозгласила своей главной целью создание к 2050-му году команды гуманоидных роботов, которые смогли бы обыграть в футбол команду чемпионов-людей. Ученые и инженеры

верят в успех этого фантастического проекта. Однако для этого потребуются огромные усилия по дальнейшему развитию конструкции и средств управления движением и поведением роботов в реальной среде во взаимодействии с другими роботами или людьми.

ГР отличаются от традиционных мобильных роботов большим числом (более 28-ми) связанных степеней подвижности, которыми необходимо управлять координировано для выполнения походки на двух ногах и манипуляций двумя руками. Он должен иметь сложное поведение. Следуя рассмотренной ранее структуре (рис.11), система управления ГР должна иметь систему осязания, интегрирующую системы зрения, акустики, тактильного и силомоментного осязания, внутреннего состояния робота и др., и специальные модули слияния сенсорной информации и формирования внутренней и внешней моделей робота (модели состояния и модели мира) для использования их всеми уровнями. Нижний уровень должен обеспечивать непосредственное управление многочисленными актуаторами (приводами степеней подвижности) ГР с использованием функций инверсной кинематики. Следующий по рангу уровень системы должен осуществлять координированное управление частями тела ГР (торсом, руками, ногами и головой) при выполнении заданных движений с использованием функций кинематической координации (например, через обученные позы). На третьем уровне необходимо реализовать элементы интеллектуального поведения ГР (видение сцены, походки разного типа, маневры, работы с неживыми объектами, отработка сложных команд и программ движения и манипулирования) с использованием заготовленного набора функций индивидуальных поведений. Для организации взаимодействия ГР с другими ГР или человеком и целенаправленного безопасного функционирования в среде людей в системе управления должен быть еще один уровень (а может быть и не один), обеспечивающий выбор и реализацию подходящих к ситуации функций взаимодействующих поведений, т. е. функций, необходимых для координации совместной работы.

Бурное развитие ГР идет по пути улучшения их конструкции и усложнения поведения. Последнее особенно важно при использовании ГР в среде людей или подобных же роботов. Для таких ГР принципиально иметь поведение, подобное человеческому. Один из путей достижения этого — создание *искусственной нервной системы* (ИНС), подобной по функциям и поведению нервной системе человека [9].

Рассмотрим основные моменты, связанные с возможностью построения ИНС для ГР. Структурно в ИНС можно выделить несколько связан-

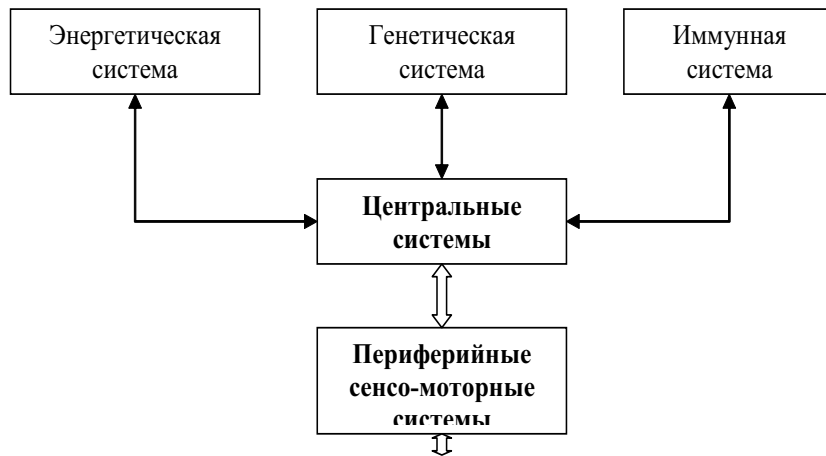


Рис. 12. Структура искусственной нервной системы

ных систем, названных по аналогии системами биологического прототипа (рис.12). Предполагается, что ИНС может быть реализована на бортовой компьютерной сети ГР. Часть аппаратных и программных средств может быть выделена на *Центральные* и *Периферийные сенсо-моторные* системы со своими наборами когнитивных и актуаторных структур соответственно. Бортовая компьютерная сеть ГР может иметь выход во внешние сети, в том числе и Интернет. Поскольку возможно проникновение компьютерных вирусов, имеет смысл говорить и об системе, защищающей бортовую сеть от вирусов и следящей за нормальным функционированием компонент. ГР является автономной машиной с собственным источником энергии, которым тоже надо управлять. Поэтому дополнительно требуется *Энергетическая* система, управляющая энергетикой робота. ИНС предполагается создавать на базе некоторого универсального набора аппаратных и программных средств, моделирующих клеточные структуры нервной системы. Для специализации и обучения этих средств функциям восприятия, действия и защиты нужна *Генетическая* система, использующая определенным образом сформированную и обновляемую генетическую информацию и процедуры самоорганизации ИНС.

Генетическая система должна иметь информацию и процедуры, необходимые для самоорганизации ИНС. Она призвана обеспечить эволюцию

ИНС при изменяющихся условиях и целевых установках робота. При «рождении» ИНС используется «генетическая» информация о структуре системы, связях компонент и процедурах, определяющих функционирование. Исходно, в начале эволюционного цикла, в ГР формируются Центральные, Периферийные, Энергетическая и Иммунная компонентные системы ИНС и определяется базовый набор внутренних агентов и их функций с фиксированными связями для их взаимодействия. Может быть определен и внутренний мета-агент, улучшающий координацию компонентных систем ИНС, но, в принципе, внутренние агенты должны уметь функционировать и без него. Внутренние агенты строятся на цепочках модулей, способных обучаться в реальном времени. В этих модулях могут быть записаны целевые установки и ограничения, а также некоторые базисные знания. Далее в процессе работы системы эти знания могут пополняться в режиме доучивания. В процессе эволюции системы модули могут клонироваться и обучаться, если требуется решение новых задач или расширяется набор функций.

Представляется возможным создание таким образом построенной ИНС для ГР на базе описанных здесь нейробиологических средств, формализованных когнитивных методов и многоагентной технологии. Предполагается, что ИНС для ГР строится, как многоагентная когнитивная система, где каждый внутренний агент, построенный на нейробиологических средствах, отвечает за свой набор поведений, получает информацию от своих или общих сенсоров, формирует сигналы управления своими или общими актуаторами и взаимодействует с другими внутренними агентами системы для организации рационального поведения робота в реальной среде. Такая многоагентная когнитивная система, согласно принципам нервно-системной организации, должна иметь иерархическую структуру с перекрывающимися функциональными компонентами, с помощью которых организуется работа внутренних агентов, реализующих индивидуальное или коллективное поведение робота. В случае коллективной работы, ИНС, управляющая ГР, должна функционировать как агент глобальной многоагентной системы, определяющей функционирование группы ГР. Для этого данный агент-ИНС должен иметь средства для коммуникаций с агентами-ИНС, управляющими другими роботами, при выполнении работ в коллективе.

Построенная на нейробиологических компонентах многоагентная система может быть настроена на требуемые поведения путем обучения каждого внутреннего агента генетически определенному для него набору когнитивных и актуаторных функций, из которых формируются когнитивно-

актуаторные процессы, объединение которых и может рассматриваться как внутренний агент. Обмен информацией и взаимодействие внутренних агентов производится через специальные нейрологические компоненты, которые адаптируются к нужному взаимодействию при пробном решении задач. Мета-агент также адаптируется, меняя приоритеты задач и организуя обработку «внимания».

Такая система может выполнять роль ИНС, поскольку она полнофункциональна и обладает универсальностью, т. е. может быть пригодна для роботов любых применений. На современном этапе развития робототехники такие системы призваны обеспечить рациональное поведение робота в среде себе подобных (без человека в искусственной или реальной жизни) или в человеческой среде (с человеком в реальной жизни).

Далее будут рассмотрены некоторые варианты систем управления интеллектуальных роботов, в которых используются или могут быть использованы нейрологические средства, когнитивный подход и многоагентная технология.

Нейрологические системы управления для интеллектуальных роботов

Система управления антропоморфного робота

В рамках проекта АРНЭ (Антропоморфный робот организации Новая ЭРА) такой робот разрабатывается в Санкт-Петербурге, ОАО «Новая ЭРА». Проект выполняется с сентября 2001 года.

Первый этап проекта завершен в июле 2003 года созданием робота АРНЭ-02 с 28-ю степенями подвижности, высотой 123 см и весом 53 кг. Базовая (актуаторная) часть системы управления реализована на бортовой сети микроконтроллеров. Дополнительные возможности управления (взаимодействие с оператором и функционирование в реальной среде с людьми) обеспечивается интеллектуальной частью системы, реализованной на удаленном компьютере. Эта удаленная часть системы принимает по радиоканалу сигналы от цветной телекамеры и микрофона, установленных на роботе. Видеосигнал обрабатывается с целью распознавания и локализации цветных реперных знаков, установленных на рабочей сцене, и некоторых объектов простой формы. Эта информация используется для управления перемещениями робота и манипуляциями с объектами. Акусти-

ческий сигнал обрабатывается с целью распознавания простых голосовых команд оператора. Имеется также возможность синтеза голосовых сообщений, которые «проговаривает» робот.

На *втором этапе* проекта, реализуемом в настоящее время, разрабатывается более совершенная система управления, которая может быть использована как интеллектуальная система управления роботами или другими автономными мобильными аппаратами (беспилотными самолетами, торпедами, ракетами и пр.). Этот вариант системы управления создается на базе когнитивного подхода и комбинированной технологии, включающей применение нейробиологических средств, техники слияния сенсорной информации, техники интеллектуального управления поведением и исполнением действий (рис. 13).

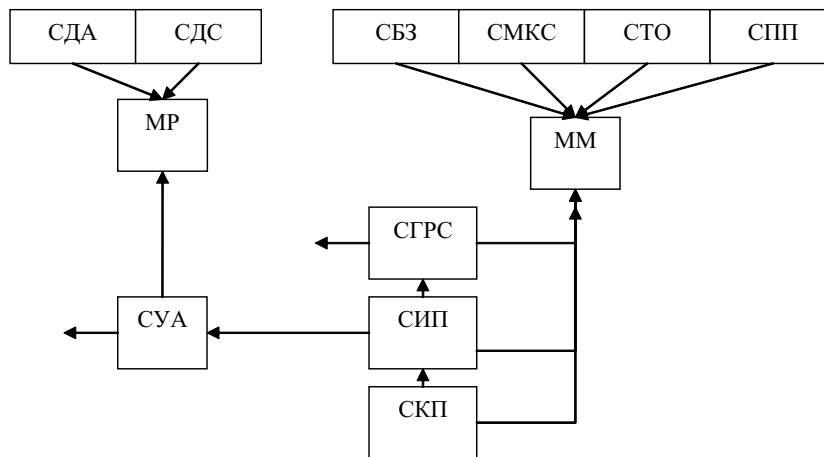


Рис. 13. Структура системы управления антропоморфным роботом

В структуре, показанной на рис. 13, использованы следующие сокращения: СДА — Система Датчиков Актуаторов, СДС — Система Диагностики Состояния, СБЗ — Система Бинокулярного Зрения, СМКС — Система Многоканального Слуха, СТО — Система Тактильного Очувствования, СПП — Система Пространственного Положения, МР — Модель Робота, ММ — Модель Мира, СУА — Система Управления Актуаторами, СГРС — Система Генерации Речевых Сообщений, СИП — Система Индивидуального Поведения, СКП — Система Коллективного Поведения.

Нейрологические средства, используемые во втором варианте системы управления, включают модули, специально разработанные для работы с когнитивными и актуаторными функциями, построенные с использованием описанных здесь адаптивных нечетко-логических моделей: проекционной, сеточной и кластерной.

Техника слияния сенсорной информации предполагает объединение результатов работы нескольких сенсорных систем робота и построение модели состояния робота и модели мира. Для нормального функционирования робота необходимы полнофункциональные наборы датчиков состояний и сенсорных систем.

Для восприятия кинематического и динамического состояний робота (взаимного положения и скоростей перемещения частей тела робота) используется **система датчиков актуаторов** (потенциометрических датчиков абсолютного положения актуаторов и токовых датчиков электроприводов степеней подвижности робота) со следующими функциями:

- определение положений терминальных точек частей робота (рук, ног, туловища, шеи и головы);
- определение статического центра масс робота;
- определение динамических параметров терминальных точек робота (скоростей и ускорений);
- определение динамического центра масс робота;
- распознавание поз робота (положение кинематически связанных частей робота).

Для восприятия диагностической информации о состоянии систем робота разработана **система диагностики** (строится на датчиках энергетики, отказов приводов, отказов электроники) со следующими функциями:

- определение уровня запаса энергии;
- определение отказов и диагностика актуаторов с электроприводами;
- определение отказов и диагностика модулей электроники;
- определение перегрузки робота при выполнении силовых действий.

Для восприятия зрительной информации о среде разрабатывается **система бинокулярного зрения** со следующим предполагаемым набором когнитивных функций и процессов:

- выделение и сегментация полутоновых объектов;
- обучение и распознавание полутоновых объектов произвольной формы (роботов, машин, людей и пр.);

- вычисление координат в трехмерном пространстве сцены методом триангуляции;
- распознавание сцены (набора объектов и локализации их на сцене);
- распознавание лиц (до 10-ти на сцене);
- распознавание артикуляции губ и лица;
- распознавание жестов-команд;
- распознавание примитивных движений с целью дальнейшего воспроизводства системой управления;
- слежение за выделенными объектами;
- слежение за изменениями на сцене (движение объектов, появление и исчезновение их и т. д.).

Для восприятия акустической информации о среде разрабатывается *система многоканального слуха* со следующим предполагаемым набором функций:

- распознавание командных предложений с распознаванием содержания команд;
- распознавание слитно произносимых предложений;
- синтаксический и семантический анализ предложений;
- формирование семантической базы знаний при обработке речевых сообщений;
- устранение влияний акустических помех с выделением источника полезного сигнала;
- распознавание голосов (до 10-ти в помещении);
- ориентация на источник знакомой речи.

Для восприятия внешних силовых воздействий разрабатывается *система тактильного осязания* со следующими функциями:

- определение формы объекта при осязании вслепую;
- определение места и величины силового воздействия.

Для пространственного восприятия разрабатывается *система пространственного положения* со следующими предполагаемыми функциями:

- получение вертикали в поле сил тяжести;
- определение динамических возмущений;
- определение курса;
- определение динамических возмущений при заданном курсе;

- определение ориентации в заданном трехмерном пространстве (курса, тангажа и крена);
- определение динамических возмущений при заданной ориентации.

Техника сенсорного слияния позволяет объединить некоторые датчиковые и сенсорные системы для их совместного использования при реализации сложного поведения робота. Так, результаты работы системы датчиков актуаторов и системы диагностики, предлагается использовать для создания системы «Внутренняя модель», которая интегрирует всю воспринятую информацию о внутреннем состоянии робота. Эта система должна вырабатывать вектор признаков текущего внутреннего состояния робота, который необходим для работы когнитивных и актуаторных структур системы управления ГР. Результаты работы систем бинокулярного зрения, многоканального слуха, тактильного и силового очувствления и пространственного положения, предлагается использовать при создании системы «Модель мира», которая интегрирует всю воспринятую этими системами информацию состоянии среды, окружающей робот. Эта система должна формировать набор знаний о среде, которые необходимы для работы когнитивных структур системы управления робота.

Техника управления поведением предполагает использование интеллектуальных методов для реализации координированного управления поведением робота в условиях неопределенности среды. Для нормального функционирования робота необходим полнофункциональный набор средств управления поведением.

Для речевого общения робота с людьми и роботами разрабатывается **система генерации речевых сообщений** со следующими предполагаемыми функциями:

- генерация сообщений из памяти в ответ на команды (ответчик);
- проговаривание текстовой информации (диктор);
- синтез сообщений в диалоге (собеседник).

Для управления роботом, функционирующим при отсутствии контактов с другими роботами, разрабатывается **система управления индивидуальным поведением** со следующими функциями:

- генерация плана действий в соответствии с заданием;
- создание в режиме обучения собственной модели объекта по интегрированной информации от датчиков и систем очувствления;
- обучение работе в соответствии со сформированной моделью;

- создание в режиме обучения модели среды (сцены) по интегрированной информации от системы бинокулярного зрения и тактильного осязания;
- навигация с использованием построенной модели среды;
- управление поведением при исполнении плана целенаправленных действий робота;
- обучение действиям, основанное на наблюдении за подобными действиями;
- обучение навыкам выполнения специальных действий (например, игре с мячом).

Для управления роботом, функционирующим в коллективе роботов или людей, разрабатывается *система управления коллективным поведением* со следующими функциями:

- координация действий при выполнении работ в группе роботов;
- координация действий при контактах с людьми;
- обучение взаимодействию при работе в группе роботов;
- обучение взаимодействию с человеком;
- обучению взаимодействию в игре (предполагается использование робота в играх на Кубок Роботов — RoboCup).

Техника управления движениями предполагает использование методов координированного управления движениями частей тела робота (рук, ног, торса, шеи и головы).

Для управления телом робота с 28-ю степенями подвижности разрабатывается *система управления актуаторами* со следующими функциями:

- синхронизация и буферизация при взаимодействии с системой управления индивидуальным поведением (формирование очереди заданий на движения и обратных связей для коррекции внутренней модели робота);
- формирование координированных траекторий конечных элементов тела робота при отработке заданий на согласованные движения (ходьба, бег, прыжки и манипуляции с разными параметрами);
- определение заданий для степеней подвижности частей тела через функции инверсной кинематики;
- коррекция заданий для обеспечения устойчивости в малом (при малых силовых возмущениях, компенсируемых за счет движения стоп);

- коррекция заданий для обеспечения устойчивости в большом (при больших силовых возмущениях, компенсируемых за счет изменений позы тела робота);
- коррекция заданий для обеспечения устойчивости при отрыве от поверхности опоры (при прыжках с использованием системы пространственного положения);
- управление отработкой заданий приводами степеней подвижности с учетом динамики.

Система управления гуманоидного робота

В перспективе антропоморфный робот АРНЭ должен трансформироваться в робот гуманоидного класса. Определяющими факторами такой трансформации являются улучшение конструкции робота (повышение гибкости, реактивности, разработка специальной энергетической установки и т. д.), а также обеспечение человекоподобного поведения робота в среде людей или себе подобных роботов. В плане второго фактора перспективным направлением является разработка системы управления для гуманоидного робота по типу искусственной нервной системы, о которой здесь уже говорилось. Поэтому архитектуру системы управления гуманоидного робота предполагается строить на основе многоагентной технологии и средств самоорганизации системы (рис. 14). При этом предполагается сохранить как базовую часть, рассмотренную ранее систему управления антропоморфного робота.

Многоагентная технология позволяет строить систему управления роботом, как набор внутренних (виртуальных) агентов, каждый из которых обеспечивает свой набор поведений, а совместная работа агентов позволяет реализовать рациональное поведение робота в текущей ситуации.

Используя такой подход, предлагается выделить следующие внутренние (виртуальные) **агенты** системы:

- агент «Слышу-Говорю», реализующий акустическое взаимодействие с объектами среды (акустическое восприятие и голосовое общение с людьми или другими роботами);
- агент «Вижу-Слышу-Говорю», реализующий зрительно-акустическое взаимодействие с объектами среды (зрительное и акустическое восприятие и общение);

- агент «Визу-Перемещаюсь», реализующий зрительное восприятие и целенаправленные перемещения среди объектов среды;
- агент «Визу-Манипулирую», реализующий зрительное восприятие и целенаправленные манипуляции объектами среды;
- агент «Слышу-Визу-Координирую», реализующий зрительно-акустическое восприятие и координированное поведение при работе в группе роботов;
- агент «Безопасно-Контактирую», реализующий контроль безопасности и коррекции действий в опасных ситуациях при взаимодействии с людьми;
- агент «Слышу-Визу-Учусь», реализующий зрительно-акустическое восприятие и обучение поведению в разных режимах.

В соответствии с многоагентной технологией и когнитивным подходом система управления ГР может быть реализована в виде коллектива из M когнитивных агентов, взаимодействующих по принципам кооперации или конкуренции. Такие агенты являются самостоятельными когнитивными системами, имеющими свои роли в коллективе агентов. Объединения агентов с разными ролями создают формации.

Пусть имеется коллектив агентов и набор ролей по числу агентов, которые могут им назначаться, т. е.:

$$CA = \{ca_1, \dots, ca_M\}; R = \{r_1, \dots, r_M\}, (r_i \neq r_j), \quad (18)$$

где ca_i — i -й агент, r_i — i -я роль. Формации являются компонентами, объединяющими роли из набора R , т. е.:

$$F = \{R, (U_1, \dots, U_K)\}; U_i \subset R: U = \{r_{i1}, \dots, r_{iK}\} (r_q \neq r_l), \quad (19)$$

где F — набор формаций, U_i — i -я формация, причем в каждой формации может быть выделена главная роль.

Отображение $CA \rightarrow R$ может не определяться априорно, если когнитивные агенты гомогенны. Если в разные моменты времени жизни системы требуется организовать разные формации, роли могут перераспределяться (гибкие роли).

В качестве *стратегий взаимодействия* целесообразно использовать: Контрактный сетевой протокол (CNP — Contract Net Protocol), Торговую стратегию (NS — Negotiation Strategy) и т. п. Для многоагентных когнитивных систем хороший эффект дает кооперативная стратегия, основанная на

динамических социальных знаниях (DSK — Dynamical Social Knowledge), объединяющую CNP и NS [3]. Эта стратегия имеет две концепции: Contract Frame (CF) и DSK-base. CF состоит из элементов:

$$CF = \{goal(d_i), CA, \alpha - cut, deadline(dl_i), \\ contractset(c_1, \dots, c_m), winners\}, \quad (20)$$

где $goal(d_i)$ — кооперативная задача для взаимодействующих агентов (цель); CA — группа агентов, участвующих в кооперативном процессе; $\alpha - cut$ — процедура выделения агентов, в которых открываются CF для цели; $deadline(dl_i)$ — срок закрытия CF ; $contractset(c_1, \dots, c_m)$ — конечное число путей достижения цели; $winners$ — один или группа победителей, которые захватывают CF .

В таком варианте кооперации каждая возможная цель описывается структурой данных, где запоминается набор контрактов, который решает целевую задачу (сценарий договоренностей). В течение процесса кооперации динамически строится DSK-база. Она используется, чтобы проводить торговлю, основанную на кооперации (NS-стратегию). Такая стратегия важна для открытых автономных многоагентных систем и может работать даже при большом числе когнитивных агентов.

Средства самоорганизации включают Модель системы и компонент Самоорганизация, которые обеспечивают автоматическое конфигурирование и настройку всей системы на решение задач управления конкретным вариантом робота.

Модель системы определяется исходно и конкретизируется под цели системы. Она должна содержать накопленную ко времени создания системы генетическую информацию о конфигурации системы, среде, поведенческих функциях и процессах, которые она должна реализовать. Эта информация используется в процессе самоорганизации всей системы (самосборки, самообучения и пр.) системами компонента *Самоорганизация*.

Процесс самоорганизации реализуется с помощью специальных средств компонента Самоорганизации. Здесь необходимы средства коррекции Модели системы для реконфигурации системы под новые цели, средства построения и коррекции Модели мира для настройки на среду, Модели поведения для настройки поведенческих функций и процессов, Модели робота для настройки актуаторных функций и процессов.

Модель мира образуется при начальном взаимодействии робота с объектами среды. Она содержит текущую и прогнозируемую статическую и

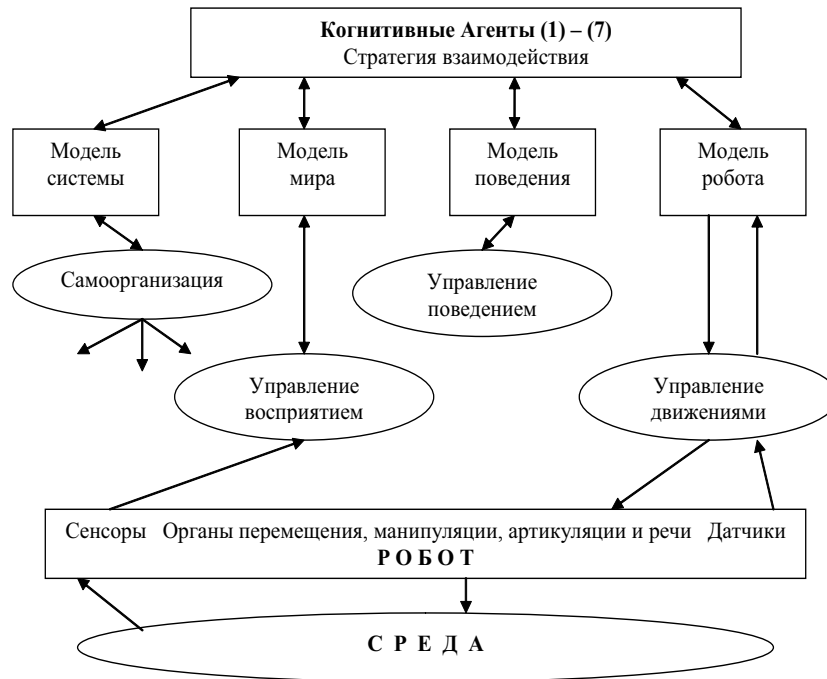


Рис. 14. Система управления гуманоидного робота

динамическую информацию о самом роботе, объектах среды и их отношениях. Такая информация в интегрированной форме получается в результате обработки многомодальной сенсорной информации системами компонента *Управление восприятием*. Интегрированная информация используется внутренними агентами системы в процессе выбора и реализации компонент поведения робота в текущей ситуации с прогнозированием последствий их реализации.

Модель поведения образуется при взаимодействиях робота с объектами среды в процессе целенаправленных действий. Она содержит набор поведений и отношений для их выбора в текущей ситуации. Эта информация используется внутренними агентами системы для организации индивидуального и коллективного поведения робота. Модель поведения используется также системами компонента *Управление поведением*, которые

исполняют поведения.

Модель робота образуется при начальном взаимодействии системы и «тела» робота. Система должна иметь изначально представление о теле робота, т. е. о его кинематике, динамике, датчиках и актуаторах, сенсорах и пр. При взаимодействии это представление конкретизируется путем настройки параметров. Эта информация о состоянии робота используется внутренними агентами системы при реализации всех управлений актуаторами робота с прогнозированием последствий их реализации. Модель робота используется системами компонента *Управление движениями*. На базе этой модели могут быть построены вспомогательные системы, обеспечивающие функционирование всей системы в целом: *Система управления энергетикой робота*, вырабатывающая действия по защите от перегрузок и подпитке источников энергии, и *Иммунная система*, защищающая программы системы управления от поражения вирусами, которые могут проникнуть через внешние каналы связи.

Рассмотренные проекты систем управления интеллектуальными роботами предполагается реализовать с использованием разработанной программной среды для конструирования когнитивных систем управления и агентов с реализацией когнитивных функций на сети нейробиологических модулей. После разработки набора когнитивных функций предполагается использовать эту программную среду для создания когнитивных и актуаторных структур, а также когнитивной системы управления и когнитивных агентов в целом при разработке описанных здесь систем управления антропоморфного и гуманоидного роботов.

Заключение

В настоящее время создан программный пакет COM Toolkit для разработки и моделирования нейробиологических средств систем управления. Он основан на фундаментальной COM (Component Object Model) технологии и написан на языке C++. В пакете имеются средства проектирования нейробиологических модулей, а также когнитивных и актуаторных структур на таких модулях с интервально-логическим, нечетко-логическим и вероятностно-логическим базисами. Все модули поддерживают грануляцию на входах. Моделирование показало, что наиболее предпочтительными для решения многих задач управления с адаптацией в реальном времени являются нейробиологические модули с нечетко-логическим базисом, построенные на базе сетевой и кла-

стерной моделей. Разрабатываются также специальные обучаемые модули, которые более эффективны при использовании обучения с подкреплением.

Разработка и исследование антропоморфного робота АРНЭ с системой управления, построенной для отработки основных конструктивных и алгоритмических решений, показали необходимость использования обучаемых компонентов. Моделирование актуаторных и когнитивных структур, построенных на нейрологических средствах, в плане разработки перспективных вариантов систем управления робота АРНЭ показало их высокую эффективность. Так, на актуаторных уровнях нейрологические компоненты позволяют, например, решать в реальном времени задачи инверсной кинематики при управлении позами робота с одновременным изменением до 22-х степеней подвижности. Использование нейрологических средств позволяет, именно за счет обучения, значительно лучше решать задачи обеспечения устойчивости робота при больших возмущениях, а также оптимизации походки по критерию минимума расхода энергии. Применение нейрологических средств на верхних уровнях систем управления позволяет строить когнитивные структуры, обеспечивающие сложное поведение роботов, например, при обходе нескольких движущихся объектов.

Проведено моделирование с использованием нейрологических средств некоторых когнитивных структур перспективной системы управления гуманоидного робота. К ним относятся и когнитивные структуры, обеспечивающие работу ряда внутренних агентов системы, например, защиту от отказов, управление энергетикой, а также выполнение целевых задач (навигацию в среде с движущимися препятствиями, игру с мячом, собирание объектов пр.). Поведение роботов гуманоидного класса отрабатывается и исследуется на примере управления виртуальным роботом-футболистом в симуляционной среде соревнований RoboCup Soccer, проводимых с 1997 года среди университетов и фирм, занимающихся разработкой роботов игровых и промышленных применений. На основе этих исследований отрабатывается программное обеспечение гуманоидных роботов, позволяющее им играть в футбол в команде себе подобных и выступать в гуманоидной лиге RoboCup. Игровое применение такого робота позволит отработать управление сложными движениями и поведением в группе, что необходимо для его нормального использования в реальной среде с участием людей или других роботов.

Исследования программных агентов в различных виртуальных средах показали хороший эффект от использования нейрологических средств в агентах-спасателях (игровая среда RoboCup Rescue), а также в агентах-

пилотах (среда воздушного симулятора для отработки взаимодействий беспилотных летательных аппаратов при выполнении заданий в группе).

Литература

1. Станкевич Л.А. Когнитивные нейробиологические системы управления // *Проблемы нейрокибернетики* (Материалы XII Международной конференции по нейрокибернетике, Ростов-на-Дону, октябрь 1999), Ростов-на-Дону, 1999.
2. Станкевич Л.А. Многоагентные когнитивные нейробиологические системы управления // IV Всероссийская конференция «*Нейрокомпьютеры и их применение*», НКП2000 (Москва, 16–18 февраля 2000), М., 2000.
3. Weiss G. (Ed.). Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. – The MIT Press, Cambridge, MA, London, UK, 1999.
4. Нильсон Н. Мобильный автомат, построенный с использованием принципов искусственного интеллекта // В сб. статей «*Интегральные роботы*» под ред. Г.Е. Поздняка, Мир, 1973.
5. URL: <http://androidworld.com>
6. Tevatia G. and Schaal S. Inverse Kinematics for Humanoid Robots // In Proceedings of IEEE Int. Conference on Robotics and Automation (ICRA 2000), 2000.
7. Юревич Е.И. Основы робототехники. – М.: Машиностроение, 1985.
8. Станкевич Л.А., Лытарева Ю.И. Когнитивные структуры в системах управления роботами // Материалы 13-й НТК «*Экстремальная робототехника*» (16–18 апреля 2002 г.), Изд-во СПбГТУ, СПб, 2002.
9. Станкевич Л.А. Искусственная нервная система гуманоидного робота // Труды Юбилейной международной конференции по нейрокибернетике (Ростов-на-Дону, 25–28 сентября 2002), Ростов-на-Дону, 2002.
10. Соболев П. В журнале «Компьютерра», 21 октября 2003, №40(515).
11. Albus D. Brain Behavior and Robotics. BYTE Books, 1987.
12. Zadeh L. Towards a Theory of Fuzzy Information Granulations its Centrality to Human Reasoning and Fuzzy Logic // *Fuzzy Set and Systems*, vol. 90, 1997.
13. Barto A. and Sutton R. Reinforcement Learning: An Introduction. – The MIT Press, 1998.
14. Солто Р. Когнитивная психология. – М.: Мир, 1967.
15. Шенерд Г. Нейробиология. Том 1. – М.: Мир, 1987.
16. Hayes-Roth B. An Architecture for Adaptive Intelligent Systems // *Artificial Intelligence*, vol. 72, 1995, pp. 325–365.

17. *Anderson J.* The Architecture of Cognition. – Harvard University Press, Cambridge, MA, 1983.
18. *Laird J. et al.* Soar: An Architecture for general intelligence // *Artificial Intelligence*, **33**(1): 1–64, 1987.
19. *Bratman M.* Intention, Plans, and Practical Reasoning. – Harvard University Press: Cambridge, MA, 1987.
20. *Котенко И. В., Станкевич Л. А.* Командная работа агентов в реальном времени // *Новости искусственного интеллекта*, №3 (57), 2003.

Лев Александрович СТАНКЕВИЧ, кандидат технических наук, профессор Санкт-Петербургского политехнического университета, начальник лаборатории моделирования и управления сектора робототехники ОАО «Новая ЭРА». Научные интересы — интеллектуальное управление, когнитивные системы, нейробиологические средства, многоагентные системы управления, футбол роботов, гуманоидные роботы. Автор более 100 печатных работ, 4 учебных пособий и 1 монографии. Организатор команд по футболу роботов и участник чемпионатов мира по футболу роботов (RoboCup) в симуляционной лиге. Является одним из организаторов создания первого в России гуманоидного робота АРНЕ.

С. А. ТЕРЕХОВ

Лаборатория искусственных нейронных сетей ALIFE,

Троицк, Московская обл.

E-mail: alife@narod.ru

**НЕЙРО-ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ
АВТОНОМНЫХ АГЕНТОВ**

Аннотация

В лекции рассматриваются задачи многошаговой оптимизации и управления сложными динамическими системами. Обсуждаются постановки типовых задач и особенности алгоритмов нейро-динамического программирования, использующих для аппроксимаций функций оптимального поведения аппарат искусственных нейронных сетей. Показана связь задачи целевой адаптации автономного агента при обучении с подкреплением и задачи прогностического оптимального управления. Приведены примеры приложений методов нейро-динамического программирования.

S. A. TEREKHOFF

Artificial Neural Network Laboratory (ALIFE),

Troitsk, the Moscow Region

E-mail: alife@narod.ru

**NEURO-DYNAMICAL PROGRAMMING FOR AUTONOMOUS
AGENTS**

Abstract

This lecture is devoted to multistage optimization methods in the control of complex dynamical systems. Statements of typical control tasks are discussed. The key technology — neuro-dynamical programming method — is based on the approximation of cost-to-go function by artificial neural networks. Examples show relations between reinforced adaptation of autonomous agents and predictive control problems.

Введение: Примеры систем с динамически-оптимальным управлением

Рациональная деятельность по ведению хозяйства, использованию ресурсов, оптимальной организации деловой активности является основной отличительной чертой устройства современной экономики и управления. Формальная сторона проблемы сводится к определению в некотором смысле наилучшего, оптимального, варианта решения в текущих условиях. В последние десятилетия сформулированы практически важные постановки задач поиска оптимума, а также предложены базовые алгоритмы для их решения. Наибольший прогресс в математических методах достигнут при рассмотрении статических задач, в которых решение принимается «одномоментно», а оптимизируемой функцией является оценка последствий выполнения такого решения. Однако, в условиях ускоряющегося темпа деловой активности и усложнения режимов конкуренции лицо, принимающее решения, все чаще сталкивается с необходимостью поиска оптимума в многошаговых задачах. Предметом рассмотрения в данной лекции являются системы, в которых решение принимается поэтапно. При этом оптимальным должен быть весь многоэтапный процесс смены состояний системы. Приведем примеры многоэтапных задач.

1. *Менеджмент проекта.* Управление ресурсами производственного или исследовательского проекта предполагает принятие серии последовательных решений. Способ организации управления на основе многошаговых решений называют “real options”.
2. *Управление оптовыми закупками в торговых сетях* (супермаркетах, специализированных магазинах), сетях быстрого питания, сетях косметических салонов и др. Определение объема и сроков поставок партий товаров на каждый момент времени должно учитывать будущие потребности и дифференцированные риски от нехватки или переизбытка товара. Управляющие решения (закупки) проводятся периодически и процесс их оптимальной выработки является многошаговым.
3. *Управление динамикой инвестиционного портфеля.* Регулярное участие в торгах и реструктуризация состава активов портфеля позволяют рассматривать задачу управления им как многошаговую. Управляющие воздействия состоят в покупке и продаже некоторых активов портфеля с целью достижения выбранного критерия эффективности

(например, обеспечение максимальной доходности на указанном масштабе времени, при заданном уровне риска).

4. *Динамическое планирование загрузки каналов сети сотовой связи* [1]. Цель состоит в обеспечении качественного сервиса для максимального числа пользователей при заданном диапазоне выделенных частот. Сложность состоит в необходимости резервирования части ресурсов для обеспечения непрерывности связи при перемещении абонента из зоны одной соты к другой. При этом слишком большой резерв равносителен большому числу отказов новым абонентам при новых вызовах (т. е. блокировок). Задача решается многократно, в реальном времени, с учетом текущей и прогнозируемой динамики запросов к сети.
5. *Поиск информации в распределенной сети*. Маршрутизация запросов в распределенном хранилище информации представляет собой задачу динамической оптимизации. Здесь потенциальный выигрыш от расширения масштаба поиска в сети противоречит готовности выполнять новые поисковые запросы. Одновременно с задачей поиска может решаться задача динамического оптимального размещения информации в сети.
6. *Динамическое составление расписаний*. Одно из актуальных применений — динамическое назначение пилотов на рейсы в крупных авиакомпаниях. Задача содержит сложные динамические ограничения, вызванные фактическим выполнением воздушных перевозок.
7. *Широкий круг «проблемных» задач*, таких, как обучение в многошаговых играх, адаптивное управление в распределенных и автономных системах, активное обучение (обучение агента, сопровождающееся активным отбором информации для дальнейшего обучения), адаптация робота в сложной окружающей среде.

Все перечисленные задачи обладают одной общей фундаментальной сложностью — оптимальный на данном шаге вклад в функцию ценности может соответствовать решению, далекому от оптимума для всего многошагового процесса в целом. Особенно четко это наблюдается в прототипических задачах поиска выхода из лабиринта: глобально оптимальное решение включает локальные отрезки, на которых происходит фактическое удаление от точки выхода. Следствием этого феномена в общественной жизни являются, например, факты «непопулярных» и, на первый взгляд, неоптимальных решений со стороны правительств, при выходе их стран из кризисов. Одним из современных подходов к многошаговым проблемам служат при-

ближенные методы поиска адаптационной стратегии, включающие аппроксимации функций оптимального поведения искусственными нейронными сетями. Общее семейство таких нейросетевых методов получило специальное название — *нейро-динамическое программирование*. Исследователи в области адаптации автономных агентских систем [2] в сложно-устроенном окружении заинтересованы в методах нейро-динамического программирования, поскольку эти методы не требуют построения прямых моделей среды и позволяют агентам начинать действовать, не дожидаясь завершения обучения.

Дальнейшее изложение лекции построено по следующему плану. Вначале мы кратко остановимся на классической постановке задачи оптимизации с ограничениями и перейдем к задаче оптимального управления системой, модель которой известна. Далее будет рассмотрена задача управления в условиях, когда модель системы недоступна. Затем мы откажемся от требования детерминированности системы и перейдем к марковскому процессу принятия решения для агента во внешней среде. Будут сформулированы постановки задач динамического программирования и уравнение Беллмана. Оставшаяся часть лекции обзора методов нейро-динамического программирования. Отдельно особое внимание будут обращено отличию задачи обучения нейронной сети на основе локальных подкреплений (reinforcement learning) от обычной задачи обучения с учителем. В завершение будут более подробно рассмотрены некоторые приложения.

Классическая задача оптимизации с ограничениями

Пусть исследуемая система описывается набором (вектором) переменных $x = \{s, a\}$, включающим переменные состояния и переменные, описывающие возможные целенаправленные изменения (управления) в системе. Пусть далее определена функция ценности (цели) $F(x)$, которую желательно максимизировать. В классической постановке эта оптимизация целиком выполняется во временной точке, без учета характера последующей динамики системы. В результате применения управлений система сразу переходит в окончательное состояние $s(a)$, которое и оценивается целевой функцией. Задача оптимизации может формулироваться, например, следующим способом¹:

¹Литература по методам оптимизации крайне обширна. См. напр., [3–5].

Найти a^* , такое, что

$$a^* = \arg \max_{a \in A} F[s(a), a],$$

где A — множество допустимых управлений, определяемое ограничениями

$$g[s(a), a] = 0,$$

$$f[s(a), a] \leq 0.$$

Постановки задач такого типа оказываются весьма содержательными на практике. Достаточно отметить, что даже «простейший» вариант этой задачи, в котором все функции предполагаются линейными, достаточен для формулирования большинства предметных задач исследования операций. Обзор тематики задач оптимизации выходит за рамки данной лекции. Качественно представление о современном состоянии в этой области можно получить на сайте optimization-online.org. При описании задачи оптимизации предполагалось, что система мгновенно переходит в конечное состояние исключительно вследствие управляющих воздействий. Для задач, в которых существенна динамика системы, это является серьезным упрощением. Отказ от него приводит к проблеме поиска оптимальных динамических траекторий, которая относится к классу задач оптимального управления.

Управляемая оптимальная динамика

Поиск оптимального управления находится на следующей ступеньке по сложности, так как в задаче оптимизации неизвестным являлся конечный набор чисел — оптимальных значений переменных, а здесь мы имеем дело с неизвестной *функцией* — законом управления. Поэтому иногда задачу управления называют задачей оптимизации в «бесконечномерном» пространстве.

Программное управление

Относительно просто может быть сформулирована так называемая *задача программного управления*. В ней предполагается, что управляющие сигналы воздействуют на систему на всем протяжении процесса, вплоть до достижения конечного состояния, однако эти сигналы не зависят от фактической траектории управляемой системы. Автомат программируется на

следование некоторой, заранее синтезированной, программе, которая выполняется независимо от того, что происходит с системой. Другими словами, система «не наблюдается» агентом-автоматом. Математически задача программного управления формулируется как задача поиска минимума функционала, который часто предполагается аддитивным:

Найти $a^(t)$, такое, что*

$$\int_0^T F[s^*(t), a^*(t)] dt + F_{t=T}[s^*(T)] \Rightarrow \max,$$

где $a^ \in A$ — множество допустимых управлений, определяемое ограничениями на всей траектории*

$$g[s(a(t)), a(t)] = 0,$$

$$f[s(a(t)), a(t)] \leq 0.$$

В выражении для оптимизируемого функционала выделен член, отвечающий оценке конечного (терминального) состояния системы. В приложениях [6] рассматриваются также различные упрощения в постановке этой задачи (например, квадратичные функционалы и линейные ограничения).

Отметим важную специфику задачи оптимального управления, которая сохранится в дальнейшем при постановке других задач на протяжении всей лекции. Изменение функции управления влечет за собой изменение траектории системы $s(a(t))$, поэтому и оптимальная траектория, и оптимальное управление должны определяться «одновременно». Итерационное варьирование траектории или управления при поиске оптимума может быть затруднено, поскольку в отсутствие решения не ясно, какие вариации являются допустимыми.

Для численного решения задачи программного управления класс управлений ограничивают параметрическим семейством функций и сводят задачу к конечномерной задаче поиска оптимальных значений параметров (т. е. к уже рассмотренной задаче оптимизации). Часто ограничиваются выбором кусочно-постоянных управлений или гладкими сплайнами.

Представляет большой интерес возможность использования нейронной сети в качестве пробной функции, так как она обладает свойствами контролируемой гладкости и ограниченными производными.

Прогностическое управление на основе нейросетевой модели системы

Программное управление является приемлемым подходом во многих прикладных ситуациях. На этом принципе основаны, например, простые металлорежущие станки или бытовые стиральные машины. В этих случаях система полностью подчинена управлению².

В более сложных случаях поведение системы под воздействием программного управления может оказаться *неустойчивым* и для его стабилизации необходима возможность наблюдения за системой (измерения состояния) и использования этой *обратной связи* в законе управления. Оптимальное *управление с обратной связью* так же, как и задачи оптимизации, имеет свою глубокую историю и многообразие постановок задач и методик их решения [7].

Здесь будет рассмотрен только один из подходов этого класса — *прогностическое управление на основе модели* (MPC — Model Predictive Control). Метод MPC особенно ценен для тематики этой лекции по трем основным причинам:

- Метод основан в явной форме на многошаговом поиске решения.
- В качестве базового инструмента может использоваться искусственная нейронная сеть, что повышает технологичность подхода.
- Метод разработан до прямых промышленных приложений с большим коммерческим эффектом.

Алгоритмы MPC используют численную оптимизацию для нахождения оптимального управления на некотором временном горизонте в будущем, основываясь на модели процесса. В современных приложениях, таких, как очистка нефти, нефтехимия, целлюлозная и пищевая промышленность, отклик управляемой системы часто является существенно нелинейным. Это, наряду с более высокими ограничениями на качество продукта и увеличивающимися требованиями производительности, экологическими требованиями и, не в последнюю очередь, экономическими соображениями требует, чтобы система эксплуатировалась в режимах, близких к технологически допустимым границам. Это, в свою очередь, требует разработки точных моделей не только для отклика системы, но и для ограничений.

Нейронные сети де-факто стали очень популярной методикой для моделирования нелинейной многомерной динамики систем, вследствие их

²В более современных устройствах имеются аварийные блокировки — пример управления по обратной связи.

уникальных аппроксимационных свойств [8]³.

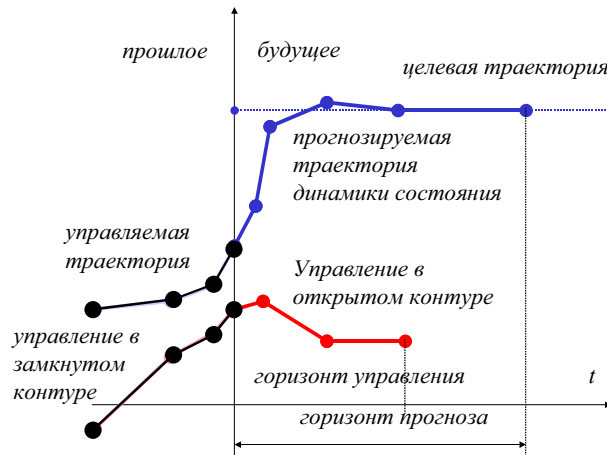


Рис. 1. Прогностическое управление на основе модели динамики системы

MPC включает следующие шаги:

- Нейронная сеть обучается на исторических данных и данных, полученных в результате испытаний управляемой системы в условиях программного управления. После обучения нейронная сеть может использоваться для предсказания динамики системы при заданном законе управления на N последовательных шагов по времени (горизонт прогнозирования).
- На каждом шаге управления новое состояние системы измеряется в петле обратной связи.

³Более основательное и, скорее всего, более доступное изложение этих вопросов можно найти в статьях А. Н. Горбая: Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // *Сибирский журнал вычислительной математики*. – 1998. – Том 1, № 1. – С. 11–24; Обобщенная аппроксимационная теорема и точное представление многочленов от нескольких переменных суперпозициями многочленов от одного переменного // *Известия высших учебных заведений. Математика*. – 1998. – № 5 (432). – С. 6–9. – Прим. ред.

- Управляющие воздействия на последующих M шагах рассматриваются как неизвестные переменные⁴. Набор этих переменных вместе с нейросетевой моделью отклика системы синтезируют теоретическую траекторию изменения состояния системы. Оптимальные значения управляющих переменных далее определяются итерационно путем минимизацией стоимости прогнозируемой траектории.
- Только *первое* управляющее действие (из N последовательных значений управления) фактически применяется к управляемой системе. Далее алгоритм тогда повторяется для следующего шага.

В рамках МРС-подхода могут быть рассмотрены две важные прикладные проблемы управления:

- «Адаптивное» управление, при котором состояние управляемой системы следует заданной траектории. Целевая функция штрафует отклонение теоретической траектории от желательной траектории.
- «Оптимальное» управление — нахождение траектории системы, ведущей к указанному множеству конечных состояний, вдоль которой стоимость управления минимальна. Стоимость может включать как прямые материальные затраты, так и косвенные требования к управлению (технологическое качество кривой управления, стоимость модификации управляющей системы, робастность и т. д.).

В работе [9] приведено описание одной из промышленных разработок — МРС-контроллера фирмы Pavilion Technologies (США). Особенно успешны применения таких контроллеров в задаче оптимального управления химическими реакторами по получению полипропилена и других полимеров. Контроллер применяется для решения еще примерно 50 других прикладных задач.

МРС-подход не свободен от недостатков. В частности, могут возникать проблемы с устойчивостью, вызванные конечностью горизонта управления. Разумеется, они стоят менее остро, чем при применении обычных линейных систем управления к существенно нелинейной системе, работающей в нескольких операционных режимах.

Требуется также специальное обоснование адекватности нейросетевой модели в условиях нехватки обучающих данных о системе. Однако, для

⁴Обычно горизонт поиска управления M выбирается меньше горизонта прогнозирования N с целью упрощения решения задачи оптимизации. За горизонтом управления полагается постоянным.

этой проблемы в современной нейроинформатике уже накоплен мощный арсенал средств.

Нейро-динамическое программирование

Применение описанного в предыдущем разделе подхода к оптимизации динамики системы ограничено несколькими обстоятельствами:

- Необходимость наличия адекватной математической модели управляемого объекта. В некоторых важных приложениях математической модели системы может и не быть, либо ее использование требует неприемлемых вычислительных ресурсов. Для наблюдения может быть доступна лишь локальная оценка управляющего решения, так называемое *reinforcement* или «подкрепление на шаге». Особенно ярко эта ситуация проявляется в автономных системах (роботы, автопилоты и т. д.). Поиск оптимального управления в них сопровождается экспериментальным исследованием окружения. При рассмотрении таких систем лучше говорить о задаче оптимальной адаптации.
- Поиск многошагового оптимального управления путем прямой оптимизации возможен лишь в задачах с небольшим числом переменных и временных шагов. При увеличении числа состояний и возможных управлений сложность задачи оптимизации резко возрастает. Это явление названо Р. Беллманом «проклятием размерности»⁵ [10].
- В общем случае объект управления демонстрирует вероятностное поведение, которое может быть вызвано как неполнотой описания сложной системы, так и объективными характером процессов. Возможность использования вероятностного описания сложной системы и соответствующий математический аппарат — *нейросетевые аппроксимации плотности вероятности* — обсуждались в одной из предыдущих лекций автора [11]. Для представления вероятности перехода из состояния в состояние при заданном управлении могут использоваться *байесовы сети* (см., например, лекцию [12]).

Заметим, что в подходе МРС задача оптимизации на каждом этапе решается заново, найденные оптимальные управления для прошлых состояний

⁵Понятие, введенное Р. Беллманом для описания катастрофического роста числа состояний при масштабировании многомерной задачи.

системы никак не используются. Выдающейся заслугой *Р. Беллмана* явилось открытие способа решения многошаговой задачи оптимизации, при котором:

- эффективно используются ранее полученные оценки состояний системы;
- большая задача разбита на серию более простых одношаговых задач, решаемых на каждом этапе.

Метод получил название «*динамического программирования*» [10]. Более подробное изложение его приведено в следующем разделе, а здесь мы проиллюстрируем основную идею на простом примере максимизации многомерной аддитивной функции с ограничениями, предложенном самим *Р. Беллманом*.

Решить задачу оптимизации

$$F(x_1, x_2, \dots, x_N) = \sum_{i=1}^N g_i(x_i) \Rightarrow \max,$$

при ограничениях

$$x_i \geq 0, \quad \sum_{i=1}^N x_i = c.$$

Здесь функции g непрерывны при неотрицательных x . Так как максимум F зависит только от числа N элементов в сумме (при сохранении порядка) и величины c , обозначим его как $f_N(c)$. Легко заметить, что для набора введенных функций f справедливо рекуррентное соотношение:

$$f_N(c) = \max_{0 \leq x \leq c} [g_N(x) + f_{N-1}(c - x)],$$

поскольку, по определению, величина $f_{N-1}(\cdot)$ описывает оптимальное значение для суммы из $(N - 1)$ упорядоченных слагаемых, а аргументы x и $(c - x)$ гарантируют выполнение ограничений задачи. Рекурсия замыкается соотношением

$$f_1(c) = g_1(c).$$

Таким образом, исходная многомерная задача оптимизации свелась к последовательности одномерных задач, которые можно рассматривать как многошаговый процесс поиска оптимума. В некоторых приложениях индексу N придается смысл дискретного времени.

Марковский процесс принятия решения

Рассмотренная выше многошаговая задача оптимизации может быть положена в основу более широкого класса задач оптимального управления и поведения. До этого момента в лекции неявно предполагалось, что управляемая система представляет собой некоторый технический объект, который рационально используется путем применения к нему управляющих воздействий. Оптимальное управление может быть специально синтезировано для этой системы, экспериментально отработано и внедрено. Модификация системы управления в технических приложениях — относительно редкий процесс, который рассматривается скорее как исключение, нежели как правило. В дальнейшем изложении мы перейдем к другой метафоре, которая служит мостиком от теории оптимального управления к теории адаптивного обучения в изменяющихся внешних условиях. Примером служит построение оптимальных стратегий поведения *автономного агента*, самостоятельно принимающего решения.

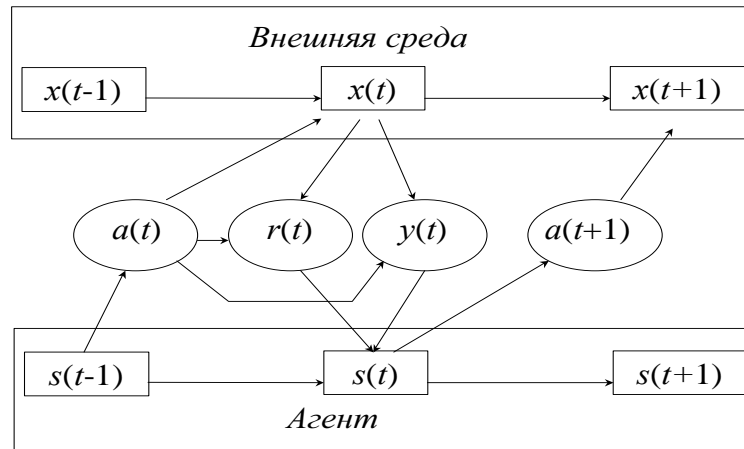


Рис. 2. Структура взаимодействия агента со средой

Агент находится в недетерминированной среде, модель которой ему, вообще говоря, недоступна. Эта задача в сравнении с рассмотренными ранее в лекции постановками еще на ступеньку сложнее. Чтобы перейти к

ней введем сначала основные понятия, сопровождающие процесс принятия решения.

Под *агентом* будем понимать автономную сущность (компьютерную программу, робототехническую систему или некоторую формальную систему — предметный смысл может уточняться в контексте), которая имеет возможность получать информацию о своем окружении и действовать, изменяя состояние (свое и окружения). Общая структура взаимодействия агента и окружения приведена на рис. 2.

Все временные изменения происходят в дискретные моменты времени⁶. *Внешняя среда* описывается состоянием среды $x(t)$, которое вероятностным образом изменяется во времени. Будем предполагать, что функция плотности распределения условной вероятности зависит только от состояния и действия агента в предыдущий момент времени:

$$x(t) \sim P[x(t)|x(t-1), a(t-1)].$$

В общем случае агент не может наблюдать все компоненты состояния среды (например, в задаче ориентации в лабиринте агент наблюдает лишь непосредственное окружение).

Наблюдаемая функция $y(t)$ также описывается вероятностной моделью:

$$y(t) \sim P[y(t)|x(t), a(t)].$$

Кроме наблюдения $y(t)$ агент получает *локальное подкрепление*⁷ (reward)

$$r(t) = E[R(t)|x(t), a(t)].$$

То, что агент наблюдает, зависит от того, что он делает. Таким образом, учитывается, что восприятие агента является *активным процессом*.

Сам агент обладает *памятью* о своем текущем состоянии. Новое состояние $s(t)$ он волен менять по некоторому закону:

$$s(t) = f[s(t-1), y(t), r(t), a(t)].$$

В новом состоянии агент вырабатывает *управляющее действие*, придерживаясь своей *стратегии* $\pi(s)$:

$$a(t) = \pi[s(t)].$$

⁶Рассматриваются также и континуальные постановки задач, обычно это приводит к серьезным математическим усложнениям.

⁷Или наказание...

Целью агента является максимизация суммарного поощрения в процессе адаптации в среде путем научения некоторой оптимальной стратегии и функции изменения состояния:

$$J(x, a) = E \left[\sum_t \gamma^t r(t) \right] \Rightarrow \max.$$

В последнем выражении использована дисконтированная функция поощрения, что гарантирует конечность суммы даже в случае, если агент адаптируется неопределенно долго. Возможна также другая постановка, в которой в системе имеются выделенные, достижимые с конечной вероятностью *терминальные* состояния, и процесс взаимодействия со средой разбивается на отдельные временные эпохи.

Хорошей метафорой для класса рассматриваемых задач может служить задача адаптации выделенного агента в лабиринте, в котором также действуют другие агенты, со своими целями. Агент должен найти выход из лабиринта, не столкнувшись по пути с другими агентами. «Клеточный мир» лабиринта стационарен, однако наличие множества участников, перемещающихся в лабиринте по неизвестным нашему агенту законам, делает задачу его перемещения вероятностной и нестационарной.

На языке дискретных клеточных миров (и в пространстве, и во времени) может формулироваться весьма широкий класс задач логистики, планирования, и исследования операций⁸.

Для простоты, будем рассматривать ситуацию, когда взаимодействие агента со средой разбивается на множество повторяющихся эпизодов. Часто задачу ограничивают специальным случаем полной наблюдаемости $y(t) = x(t)$, как, например, в некоторых игровых постановках (поле игры доступно полностью). Тогда необходимость в выделении собственного состояния агента отпадает, и можно использовать термин «состояние» для описания состояния совокупной системы «агент + среда».

Агент должен научиться оптимальному поведению только на основе элементарных поощрений, возникающих в актах взаимодействия со средой, без внешнего учителя и без возможности непосредственного наблюдения отдаленных последствий своих действий. Другими словами, агент имеет представление о том, *что* нужно сделать (*что* будет оценено высо-

⁸Язык клеточных миров оказывается еще более широким, поскольку на нем могут рассматриваться задачи оптимизации для двух и более агентов, относящиеся к *теории игр*.

ким уровнем поощрения), но не *как* этого достичь. Это является принципиальным моментом в проблеме, и порождает следующую проблему выбора (дилемму рациональности-иррациональности):

- В процессе поиска оптимума агент должен действовать максимально целесообразно в соответствии с текущим пониманием ситуации.
- Вместе с тем, агент должен исследовать совершенно новые варианты поведения, в противном случае он остановится в локальном минимуме.

В случае, когда состояние системы содержит всю существенную информацию для описания дальнейшей динамики, задача сводится к *марковскому процессу*⁹, в котором вероятности переходов зависят от управляющих переменных (действий агента). Более строго, вероятности переходов между состояниями должны удовлетворять условию:

$$P[s(t+1) = s', r(t+1) = r | s(t), a(t), s(t-1), a(t-1), \dots, s(0), a(0)] = \\ = P[s(t+1) = s', r(t+1) = r | s(t), a(t)]$$

для всех состояний и историй системы. На практике, требование марковости часто можно выдержать, если расширить размерность пространства состояний путем введения дополнительной памяти об истории динамики системы.

Процессы, в которых вероятности переходов зависят от управляющих переменных, принято называть *марковскими процессами решений* (MDP — Markov Decision Process). Далее будет показано, что нахождение оптимальных стратегий для таких процессов может описываться компактными уравнениями, решаемыми итерационными алгоритмами.

Уравнение Беллмана

Введем обозначения. *Вероятности переходов между состояниями* будем описывать как

$$T(s, a, s') = P[S(t+1) = s' | S(t) = s, A(t) = a].$$

⁹*Марковский процесс* – процесс без последствия, т.е. случайный процесс, эволюция которого после любого заданного момента времени t не зависит от эволюции, предшествовавшей t , при условии, что значение процесса в момент t фиксировано. Другими словами, «будущее» и «прошлое» такого процесса не зависят друг от друга при известном «настоящем». — Прим. ред.

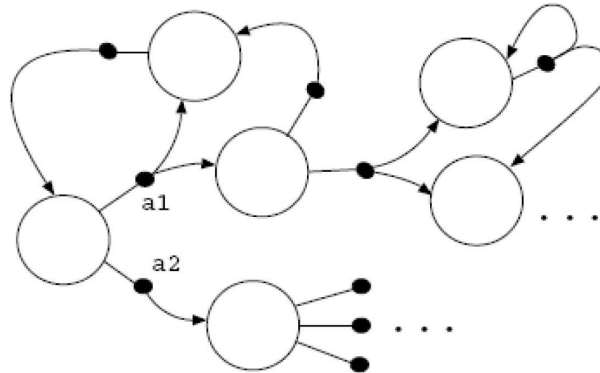


Рис. 3. Граф марковского процесса решений
Кружки соответствуют состояниям процесса, точки — возможным решениям.

В случае детерминированных переходов будем говорить о модели смены состояний $s' = f(s, a)$.

Выражение

$$r(s, a, s') = E[R(t+1) | S(t) = s, A(t) = a, S(t+1) = s']$$

описывает *функцию поощрения*¹⁰ на шаге.

Обозначим далее как $V^\pi(s)$ *цену состояния* s — т. е. то суммарное поощрение, которое получит агент, следующий своей стратегии π , в эпизоде, начинающемся в состоянии s . Заметим, что цена состояния существенно зависит от стратегии:

$$V^\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(t+k+1) | S(t) = s \right].$$

Максимально возможная цена состояния

$$V^*(s) = \max_{\pi} V^\pi(s)$$

¹⁰Поскольку целью агента является максимум математического ожидания суммы поощрений, удобно перейти от полного вероятностного рассмотрения стохастического процесса поощрения к описанию достаточной статистики — матожидания поощрения на шаге.

соответствует оптимальной стратегии π^* . Аналогично цене состояния введем также *цену действия* $Q^\pi(s, a)$ — суммарное поощрение агента, принявшего в состоянии s управляющее решение a , и далее следующего избранной стратегии

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(t+k+1) | S(t) = s, A(t) = a \right].$$

С переходом от представления в виде оценки состояния к цене хода в данном состоянии задача оптимизации на каждом шаге упрощается, однако число переменных задачи возрастает.

Цены состояния и действия связаны соотношением, вытекающим из определений:

$$Q^\pi(s, a) = \sum_{s'} T(s, a, s') \cdot [r(s, a, s') + \gamma V^\pi(s')].$$

Оптимальная стратегия соответствует максимальному поощрению:

$$V^*(s) = \max_a Q^*(s, a) = \max_a \sum_{s'} T(s, a, s') [r(s, a, s') + \gamma V^*(s')].$$

Аналогичное соотношение имеет место и для цены действия:

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [r(s, a, s') + \gamma \max_{a'} Q^*(s', a')].$$

Последние два уравнения играют фундаментальную роль в теории марковских процессов решений и динамическом программировании. Они носят название *уравнений Беллмана* (для оценок состояния и оценки действия). Решение уравнения Беллмана дает оптимальное распределение цены по всему пространству состояний задачи, и тем самым, решает сразу семейство задач оптимизации, отвечающих различным начальным состояниям.

Игровая стратегия и оценка позиции

Уравнение Беллмана является наиболее общим соотношением для оценки позиции в играх, обладающих марковским свойством¹¹. Решение уравнения Беллмана, формально, дает оценку каждого состояния (т.е. игровой

¹¹Шахматы, например, обладают таким свойством с точностью до отслеживания возможности рокировки короля. Правила запрещают рокировку, если король двигался на

позиции), поэтому оптимальная стратегия игры легко определяется — на каждом шаге следует перемещаться в состояние с более высокой ценой, в качестве которой может выступать, например, вероятность выигрыша. Выбранное действие в игровых задачах часто называют *ходом* игрока.

Буквальное описание состояния как совокупности расположения всех фигур наталкивается на большие вычислительные сложности. На практике выбираются более компактные представления [13], агрегирующие большое число элементарных состояний (как-то, позиция фигуры в центре, число угроз и защит фигур, и т. д.). В свою очередь, возникают трудности при выборе представления — противоречие между подробностью описания и «проклятием размерности». Сама проблема извлечения *существенных черт* для данной задачи является крайне сложной и должна рассматриваться в совокупности с задачей оптимизации стратегии. Тем не менее, переход к оцениванию состояния, вместо необходимости перебора и оценки каждой из возможных стратегий, является существенным шагом к упрощению задачи.

Итерации в уравнениях Беллмана

Одним из прямых подходов к поиску оптимальной адаптации в (конечных и дискретных) игровых позициях служит *итерационное вычисление оценки позиции* (value iteration).

$$V_{j+1}^*(s) \leftarrow \max_a \sum_{s'} T(s, a, s') [r(s, a, s') + \gamma V_j^*(s')].$$

Сходимость алгоритма *value iteration* легко устанавливается для показателя дисконтирования γ , меньшего единицы — итерационное отображение является сжимающим. Если дисконтирование отсутствует, то для сходимости требуются дополнительные условия на матрицу вероятностей и на достижимость состояний (т. е. при неограниченных итерациях каждое состояние посещается бесконечное число раз). Обсуждение сходимости итераций в случае непрерывных переменных и управления выходит за рамки

более «ранних» ходах в партии. Однако, в случае шахмат, эту проблему легко обойти, если расширить понятие состояния, введя дополнительную логическую переменную «двигался ли король?». В игре Го ситуация более сложная вследствие правила «Ко», запрещающего повтор расположения камней на игровой доске. Поэтому набор допустимых ходов в некоторой позиции формально зависит от всей предыстории партии.

этой лекции¹².

Другой интересной задачей является *оценивание некоторой определенной стратегии* π поведения агента (policy evaluation). Уравнение для цены состояния относительно заданной стратегии имеет вид:

$$V_{j+1}^{\pi}(s) \Leftarrow \sum_{s'} T(s, \pi(s), s') [r(s, \pi(s), s') + \gamma V_j^{\pi}(s')].$$

Здесь состояние s' находится на допустимой траектории, ведущей из состояния s при применении действия $\pi(s)$ в данной стратегии. Если получена оценка состояний в выбранной стратегии, то ее можно использовать для улучшения самой стратегии (policy iteration). А именно, гарантировано более ценной стратегией будет стратегия, «жадная» по отношению к данной:

$$\pi_{j+1}(s) \Leftarrow \arg \max_a \sum_{s'} T(s, a, s') [r(s, a, s') + \gamma V_j^{\pi}(s')].$$

При разработке алгоритмов шаги оценивания стратегии и ее уточнения могут чередоваться. В дискретном случае возникающие при прямом итерировании уравнения Беллмана трудности в основном связаны с практическим обеспечением сходимости за приемлемое время вычислений. Незвестные величины цены состояния могут представляться в виде векторов, а вероятности переходов — в виде матриц. В частности, соотношения для оценивания стратегии могут рассматриваться, как система линейных уравнений¹³

$$\mathbf{V} = \mathbf{R}^{\pi} + \gamma T^{\pi} \mathbf{V},$$

а задача вычисления оценки позиции может быть сформулирована как задача решения линейных неравенств типа

$$\mathbf{V} \leq \mathbf{R} + \gamma T \mathbf{V}$$

с дополнительными требованиями максимальной оценок.

¹²Некоторые принципиально сложные в теории обучения с подкреплением вопросы рассматриваются в работе [14].

¹³Здесь и далее по тексту лекции прямые полужирные буквы в формулах обозначают векторы, составленные из элементов, обозначаемых соответствующими «обычными» буквами с индексами. — Прим. ред.

Нейросетевая аппроксимация функции ценности

Табличное представление цен действий и состояний задачи имеет естественные *ограничения по масштабируемости задачи* на большую размерность. В дискретных моделях, имеющихся в литературе, число измерений весьма редко превосходит дюжину. Выход состоит в поиске приближенных аппроксимаций функций задачи.

Имеются два важных условия, влияющих на эффективность выбираемых алгоритмов аппроксимации. Во-первых, алгоритм должен основываться на функциях, обладающих богатыми возможностями по аппроксимации, и, во-вторых, для выбранного алгоритма должны существовать масштабируемые алгоритмы выбора параметров аппроксимаций. Эти два требования часто являются противоречащими друг другу — богатый набор аппроксимирующих функций обычно связан с большим числом свободных параметров.

Искусственные нейронные сети являются весьма естественным инструментом для приближенного представления оценок состояний при росте размерности. Однако в стандартной постановке задача аппроксимации нейронной сетью рассматривается, как задача *обучения «с учителем»*, т. е. с использованием обучающих пар вида

(вектор аргументов, значение аппроксимируемой функции).

В задаче аппроксимации ценности состояния ($V(s)$ или $Q(s, a)$) «правильные» обучающие значения отсутствуют — они являются решением задачи оптимизации, которую как раз предстоит решить. Это обстоятельство вносит новый, принципиальный уровень сложности в постановку задачи, а именно, оптимальное решение задачи, содержащей приближения в оптимизируемом функционале на классе выбранных аппроксимаций *не обязано существовать*, а если решение приближенной задачи все-таки существует, оно *не обязано приближать* собой решение исходной *точной* задачи.

На практике это приводит к тому, что приходится довольствоваться субоптимальными стратегиями, представленными аппроксимациями, и эти стратегии весьма расширительно трактуются как приближенные решения задачи оптимальной адаптации (управления). Семейство приближенных алгоритмов, основанных на нейросетевых методах аппроксимации функций оценки состояний в многошаговых задачах, принято называть *нейродинамическим программированием* [15].

Рассмотрим вначале, как может быть построено обучение нейронной сети в случае задачи итерационного вычисления цены состояния. Алгоритм *value iteration* состоит в поиске неподвижной точки уравнения типа $V = g(V)$ методом простой итерации. Операция присваивания нового итерационного значения $V_{j+1} \leftarrow g(V_j)$ в случае нейросетевой аппроксимации не может быть применена непосредственно и должна быть заменена некоторой операцией изменения весовых коэффициентов нейронной сети.

Простые варианты наталкиваются на методические сложности. Например, операцию присваивания можно было бы заменить одним шагом градиентного уменьшения ошибки

$$E(\mathbf{w}) = \frac{1}{2} [V_{\mathbf{w}}(s) - g(V)]^2,$$

который приводит к следующим формулам обновления весов нейронной сети с некоторым шагом

$$\mathbf{w} \leftarrow \mathbf{w} + \eta [g(V) - V_{\mathbf{w}}(s)] \frac{\partial V_{\mathbf{w}}(s)}{\partial \mathbf{w}}.$$

Хотя эти соотношения представляются весьма естественными, имеется несколько серьезных *контрпримеров*, указывающих на отсутствие сходимости такого алгоритма даже в случае простых линейных аппроксимаций [13]. Причина кроется в потере свойства безусловной сжимаемости отображения, в котором используется аппроксимация. Несмотря на то, что в литературе было предложено много практически полезных способов стабилизации итераций [16], общий вывод таков — локальные методы *value iteration* с аппроксимацией функции ценности в общем случае *не сходятся*.

Идея решения проблемы со сходимостью состоит в использовании для вычисления функции цены состояния *глобальной* информации о траектории из состояний, которые посещались агентом. Семейство алгоритмов, основанных на этом подходе, получило название «методы разностей по времени» (temporal difference methods, $TD(\lambda)$). В современном виде теория $TD(\lambda)$ сформулирована Саттоном и Барто в [1].

В дискретном случае алгоритмы с разностями по времени обходят (частично) также и другое ограничение методов *value iteration*, а именно, они являются *online*-алгоритмами и не требуют для новых приближений посещения всех состояний. Обучение происходит отдельными траекториями (эпизодами). В простейшей трактовке, если происходит изменение в

оценке некоторого состояния, это изменение отражается и на предыдущих состояниях. Обновление оценки происходит синхронно.

Наиболее просто алгоритм формулируется в случае дискретных состояний. Предполагается, что если оценка текущего состояния после шага уточняется на величину δV , то оценка предыдущего состояния должна быть уточнена на $\lambda \delta V$, состояния перед ним — на $\lambda^2 \delta V$ и т. д. Для накопления информации об истории состояний отводится специальная память (*eligibility trace*).

Алгоритм 1: $TD(\lambda)$ — метод разностей по времени

Require: Инициализировать $\mathbf{V} \leftarrow \text{любое}$; $\mathbf{e} \leftarrow 0$

```

for all для каждого эпизода do
   $s \leftarrow \text{случайное}$ 
  while до конца эпизода do
     $s' \sim T[s, \pi(s), s']$ 
     $\delta \leftarrow r[s, \pi(s)] + \gamma V(s') - V(s)$ 
     $e(s) \leftarrow e(s) + 1$ 
     $\mathbf{V} \leftarrow \mathbf{V} + \delta \cdot \mathbf{e}$ 
     $\mathbf{e} \leftarrow \lambda \gamma \mathbf{e}$ 
     $s \leftarrow s'$ 
  end while
end for

```

В этом алгоритме используемая стратегия $\pi(s)$ может быть постоянной, в этом случае вычисляется функция оценки состояний относительно этой стратегии. Альтернативно, стратегия может быть выбрана *жадной* по отношению к текущей оценке V , в таком случае алгоритм оценивает значение оптимальной стратегии V^* . На практике сходящееся решение достигается для ε -жадных стратегий, в которых с вероятностью ε предпринимается случайное действие, а в остальных случаях — жадное (максимизирующее текущее значение V). Обычно стратегии класса ε -жадных ассоциируются с исследовательским поведением агента.

Приведенный алгоритм обобщается на вариант с нейросетевой аппроксимацией функции оценки. Шаги внутреннего цикла имеют вид

$$\begin{aligned}\delta &\Leftarrow r[s, \pi(s)] + \gamma V_{\mathbf{w}}(s') - V_{\mathbf{w}}(s), \\ \mathbf{e} &\Leftarrow \lambda \gamma \mathbf{e} + \frac{\partial V_{\mathbf{w}}(s)}{\partial \mathbf{w}}, \\ \mathbf{w} &\Leftarrow \mathbf{w} + \eta \cdot \delta \cdot \mathbf{e}.\end{aligned}$$

Заметим, что вектор «следа памяти» \mathbf{e} определен здесь для каждого параметра нейросетевой модели \mathbf{w} , а не для набора состояний (как в дискретном случае). К сожалению, точных результатов о сходимости алгоритмов $TD(\lambda)$ с аппроксимациями известно мало.

Алгоритм *SARSA* и *Q*-обучение

Часто на практике используются алгоритмы временных разностей, в которых аппроксимируется не ценность состояния $V(s)$, а цена действия агента $Q(s, a)$ в данном состоянии. Как уже отмечалось, функция Q позволяет обойтись без модели системы, так как оптимальное управление в каждом состоянии может быть найдено непосредственно:

$$a^* = \arg \max_a Q(s, a).$$

Можно выделить два варианта алгоритмов для нахождения $Q(s, a)$. В первом варианте при вычислении разности по времени используется текущее значение функции $Q(s, a)$, т. е. разности согласованы с оцениваемой стратегией (в англоязычной литературе используется термин *on-policy*).

$$\begin{aligned}Q[s(t), a(t)] &\Leftarrow Q[s(t), a(t)] + \\ &+ \alpha \{ r(t+1) + \gamma Q[s(t+1), a(t+1)] - Q[s(t), a(t)] \}.\end{aligned}$$

Алгоритм получил название *SARSA* (в соответствии с мнемоникой обозначений последовательности $s, a \rightarrow r \rightarrow s' a'$). Циклы обучения устроены по эпохам, на каждой пробной траектории величины управлений $a(s)$, $a'(s')$ выбираются ε -жадными относительно текущей оценки Q .

Второй вариант, называемый *Q-обучением* (*Q-learning*) использует при вычислении разностей по времени *максимальную* текущую оценку Q в данном состоянии, независимо от текущей стратегии. Таким образом, при

моделировании используется одна стратегия, а фактически оценивается другая (оптимальная относительно текущей оценки Q). Такие методы называют не согласованными (*off-policy*). Уравнения для обучения агента имеют вид:

$$Q[s(t), a(t)] \leftarrow Q[s(t), a(t)] + \alpha \left\{ r(t+1) + \gamma \max_{a'} Q[s(t+1), a'] - Q[s(t), a(t)] \right\}.$$

Сравнительная оценка этих методов приведена в таблице:

<i>SARSA</i>	<i>Q-обучение</i>
Обучение стратегии, наилучшей среди ε -жадных.	Обучение оптимальной стратегии
Высокая скорость обучения, так как принимаются во внимание «исследовательские» шаги (с вероятностью ε)	Высока вероятность попадания в локальный минимум.

Для обоих вариантов предложены модификации, учитывающие «следы памяти» (называемые, соответственно, $SARSA(\lambda)$ и $Q(\lambda)$) с нейросетевыми аппроксимациями.

Значительная исследовательская активность в области алгоритмов рассмотренных классов связана с поиском методов агрегации состояний в макро-состояния (существенные черты задачи), сокращающих пространство поиска задачи, а также с теоретическим обоснованием сходимости аппроксимационных алгоритмов в специальных условиях.

Примеры прикладных разработок

В завершение лекции остановимся на двух важных примерах технологий нейродинамического программирования: управление финансовым портфелем и оптимизация ресурсов в сетях сотовой связи.

Динамическое управление портфелем финансовых активов

В работе [17] описывается пример успешного использования методов многошагового обучения для задачи управления торговой системой. Рассмотрим простейший вариант, когда портфель состоит из одного актива¹⁴ с ценой z_t , и автоматическая торговая система вырабатывает оптимальную последовательность решений $F_t \in \{-1, 0, 1\}$. Три варианта решений соответствуют покупке (*long*), пассивному ожиданию и, соответственно, продаже (*short*) актива. При этом для простоты будем считать, что объем каждой сделки фиксирован. Математически речь идет об обмене одного актива портфеля на другой по текущей рыночной цене¹⁵. В момент свершения эта сделка не изменяет стоимость портфеля, однако она вредно или полезно сказывается на его цене в будущем.

Торговая система может следовать некоторой параметрической модели:

$$F_t = F\left[\theta_t; F_{-1}; \{z_t, z_{t-1}, \dots; y_t, y_{t-1}, \dots\}\right].$$

Здесь θ — обучаемые параметры, y_{t-k} — дополнительные внешние переменные в соответствующие моменты времени (например, значения биржевых индексов, цены базовых активов и т. п.). Соотношение для дохода, получаемого системой-трейдером за T шагов, имеет вид:

$$P_T = \sum_{t=1}^T R_t = v \sum_{t=1}^T \left\{ r_t^f + F_{t-1} [r_t - r_t^f] - \delta |F_t - F_{t-1}| \right\}.$$

Здесь v — объем сделки, δ — комиссия, $r_t = z_t - z_{t-1}$ — текущая доходность, величины с индексом f соответствуют доходности безрискового актива (например, государственной ценной бумаги). Возможны другие соотношения для дохода портфеля, учитывающие, например, частичное реинвестирование прибыли [18].

Цель торговой системы может формулироваться как достижение максимального дохода. На практике, в соответствии с представлениями в области финансов, принято оптимизировать величину отношения среднего дохода

¹⁴Портфелем называется *подмножество* активов (например, ценных бумаг) из *множества* активов, обращающихся на рынке. Портфель логически дополняется еще одним активом — (свободными) деньгами.

¹⁵Обычно имеются еще издержки сделки.

к его дисперсии (*отношение Шарпа*). Для целей применения методик многошаговой оптимизации в работе [17] был предложен дифференциальный вид отношения Шарпа.

Алгоритм был применен для торговли на среднемесячных значениях индекса S&P 500. В модели использовались дополнительно еще около 80 временных рядов (как макроэкономических, так и финансовых). Для стартового обучения трейдер-системы использовались исторические данные за 1950–1969 годы. В игровом режиме система проверялась на 25-летнем промежутке с 1970 по 1994 годы. Полученное соотношение Шарпа составило величину 0.83 в год. Для сравнения стратегия «купи и держи» дает на этом растущем рынке значение 0.34 в год, в качестве безрисковых активов рассматривались годовые векселя казначейства США (US Treasury Bills).

Авторы отмечают, что торговая система, основанная на алгоритмах нейро-динамического программирования, принимала весьма качественные решения в периоды нефтяного кризиса в 1974 году, финансовых проблем начала 80-х годов, рыночных поправок 1984 года и обвала рынка в 1987 году.

Динамическое резервирование каналов сети сотовой связи

В сетях сотовой связи остро стоит проблема распределения диапазонов частот для обеспечения максимального количества услуг в условиях стохастически изменяющихся потребностей. Рассмотрим модель сотовой сети.

Заданная географическая область разделяется на непересекающиеся ячейки и каждая ячейка обеспечивает услуги в пределах своих границ. Весь доступный диапазон частот разбит на каналы. Каждый канал может использоваться одновременно в нескольких ячейках, при условии, что эти ячейки географически далеки друг от друга (так, что нет интерференции). Когда пользователь запрашивает услугу, ему либо назначается один из свободных каналов, либо вызов блокируется. Также, при пресечении границ зоны, активный вызов передается от сотовой ячейки к соседней ячейке, где ему выделяется канал. Если такого канала нет, сервис прерывается.

Таким образом, критерий оптимальности должен учитывать как сокращение числа блокировок, так и уменьшение вероятности прерывания сервиса.

Многие сотовые системы используют фиксированное распределение каналов. Более эффективной является стратегия динамического распределения, которая позволяет использовать в ячейке *любой* канал, который

свободен и в данный момент не используется в соседних ячейках.

Сформулируем эту задачу в терминах многошаговой оптимизации [15]. Смена состояния происходит, когда канал освобождается (пользователь покидает ячейку) или когда новый пользователь запрашивает канал (новый вызов), либо пользователь переходит из ячейки в ячейку (поддержка вызова). Состояние в каждый момент описывается списком занятых каналов в каждой ячейке, а также типом события, вызвавшего смену состояния (вход, выход, поддержка).

Управление состоит в оптимальном переназначении каналов, когда пользователь покидает систему. Цель переупаковки — обеспечение больших возможностей по предоставлению свободных каналов, не противоречащих ограничениям на соседство ячеек. Когда пользователь входит в систему, управляющее действие состоит в выделении канала или отказе от предоставления канала, если свободных каналов нет или если выделение имеющегося канала приведет к большому числу блокировок в следующие моменты времени. Каждый случай блокировки или отказа в сервисе приводит к увеличению штрафа на временном шаге. Цель оптимизации — найти стратегию с минимальным суммарным штрафом.

Число различных состояний для этой задачи очень велико, поэтому прямое применение методов динамического программирования для практически интересных масштабов невозможно¹⁶. Однако, приближенные решения можно получить методиками, использующими сокращенное описание состояний и дифференцирование по времени. Для простоты, можно считать поддержку сервиса при переходе от ячейки к ячейке двумя событиями — уход пользователя из одной ячейки и новый вызов в другой. Кроме того, для каждой ячейки ограничимся лишь агрегированной информацией — числом свободных каналов и для каждого канала сети — количество раз, сколько этот канал используется в данный момент соседями данной ячейки. В этом случае число состояний пропорционально произведению числа ячеек и числа каналов, и задача становится обозримой.

В работе [15] приводятся результаты для пуассоновского потока заказов в сети из 50 ячеек с 70-ю каналами, при нагрузке в несколько сотен вызовов в час в каждой ячейке. Алгоритм $TD(\lambda = 0)$ устойчиво обеспечивает меньшее число блокировок, в сравнении с лучшими известными эвристическими алгоритмами динамического распределения, и заведомо превосходит алгоритмы с фиксированным распределением частот.

¹⁶Но как же быстро растет мощность и память компьютеров!

Итоги

Симбиоз нейронных сетей и методов динамического программирования для решения многошаговых задач оптимизации — широко и интенсивно развиваемая область современной прикладной математики. Основные результаты получены в течение лишь последних 10 лет. К сожалению, не для всех условий использования алгоритмов в настоящий момент имеются строгие обоснования. С другой стороны, как и в области оптимального управления, важным исследовательским инструментом является компьютерное моделирование.

Алгоритмы нейро-динамического программирования крайне привлекательны не только для традиционных «тяжелых» задач оптимизации, но и для построения простых приближенных решений в мобильных устройствах, встраиваемых и робототехнических системах.

Благодарности

Автор выражает глубокую благодарность *Ю. В. Тюменцеву* и *Н. Г. Макаренку* за плодотворные комментарии и творческое содействие.

Литература

1. *Sutton R., Barto A.* Reinforcement learning: An introduction. – MIT Press, 1998.
2. *Wolpert D. H., Tumer K.* An introduction to collective intelligence. – NASA Ames Research Center, 2000.
URL: http://ic.arc.nasa.gov/ic/projects/coin_pubs.html
3. *Дэннис Дж., Шнабель Р.* Численные методы безусловной оптимизации и решения нелинейных уравнений. – М.: Мир, 1988. – 440 с.
4. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. – М.: Мир, 1985.
5. *Лутманов С. В.* Курс лекций по методам оптимизации. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. – 368 с.
6. *Ванько В. И., Ермошина О. В., Кувыркин Г. Н.* Вариационное исчисление и оптимальное управление. – М: Изд-во МВТУ им. Н. Э. Баумана, 2001.
7. *Филлипс Ч., Харбор Р.* Системы управления с обратной связью. – М: Изд-во «Лаборатория базовых знаний», 2002.
8. *Горбань А. Н.* Обобщенная аппроксимационная теорема и свойства нейронных сетей // *Соросовский образовательный журнал*, 1998.

9. *Piche S., Keeler J., Martin G., Boe G., Johnson D., Gerules M.* Neural network based model predictive control // *NIPS'1999*. Vol. 12, Paper No. 1029.
URL: <http://nips.djvuzone.org/djvu/nips12/1029.djvu>
10. *Беллман Р.* Динамическое программирование. – М.: ИЛ, 1960.
11. *Терехов С. А.* Нейросетевые аппроксимации плотности в задачах информационного моделирования // Лекция для Школы-семинара «Современные проблемы нейроинформатики». Часть 2, Москва, МИФИ, 25–27 января 2002 года. – М.: Изд-во МИФИ, 2002.
12. *Терехов С. А.* Введение в байесовы сети // Лекция для Школы-семинара «Современные проблемы нейроинформатики». Часть 1, Москва, МИФИ, 29–31 января 2003 года. – М.: Изд-во МИФИ, 2003.
13. *Tsitsiklis J.N., Van Roy B.* Feature-based methods for large scale dynamic programming // *Machine Learning*, **22**:59–94, 1996.
14. *Sutton R. S.* Open Theoretical Questions in Reinforcement Learning. AT&T.
URL: www.cs.umass.edu/~rich
15. *Bertsekas D. P., Tsitsiklis J. N.* Neuro-dynamic programming. – Athena Scientific, Belmont, MA, 1996.
16. *Kaelbling L. P., Littman M. L., Moore A. W.* Reinforcement learning: A Survey // *Journal of AI Research (JAIR)*, Vol. 4, 1996.
URL: <http://www.cs.brown.edu/people/lpk/rl-survey/rl-survey.html>
17. *Moody J., Saffell M.* Reinforcement learning for trading // *Advances in Neural Information Processing Systems*, **11**, M. S. Kearns, S. A. Solla and D. A. Cohn, Eds. – MIT Press, Cambridge, MA, 1999.
18. *Интрилигатор М.* Математические методы оптимизации и экономическая теория. – М. Айрис Пресс, 2002. – 567 с.

Сергей Александрович ТЕРЕХОВ, кандидат физико-математических наук, заведующий лабораторией искусственных нейронных сетей ALIFE (г. Троицк, Московская обл.). Область научных интересов — анализ данных при помощи искусственных нейронных сетей, генетические алгоритмы, марковские модели, байесовы сети, методы оптимизации, моделирование сложных систем. Автор 1 монографии и более 50 научных публикаций.

Н. Г. МАКАРЕНКО

Институт математики, Алма-Ата, Казахстан

E-mail: makarenko@math.kz

**КАК ПОЛУЧИТЬ ВРЕМЕННЫЕ РЯДЫ ИЗ ГЕОМЕТРИИ И
ТОПОЛОГИИ ПРОСТРАНСТВЕННЫХ ПАТТЕРНОВ?**

**Математические аспекты анализа распределенных динамических
систем**

Аннотация

Подходы Вычислительной, Интегральной, Стохастической геометрии и Вычислительной топологии позволяют превратить геометрию и топологию изображений в скалярные временные ряды, пригодные для традиционного нелинейного анализа. Лекция представляет собой популярное введение в современные методы анализа пространственно-временных паттернов, типичных для распределенных динамических систем.

N. MAKARENKO

Institute of Mathematics, Kazakhstan, Alma-Ata

E-mail: makarenko@math.kz

**HOW ARE TIME SERIES OBTAINED FROM GEOMETRY AND
TOPOLOGY OF SPACE PATTERNS?**

Mathematical aspects of the analysis of extended dynamical systems

Abstract

The approaches applying computational, integral and stochastic geometry as well as computational topology give possibility to transform a geometrical and topological complexity of images into scalar time series appropriate for up-to-date nonlinear analysis. This lecture is a comprehensible introduction to contemporary methods of analysis of spatiotemporal patterns, which are typical for extended dynamical systems.

Введение

Поскольку я был вынужден обрывать рассмотрение каждого случая прежде, чем оно принимало узкоспециализированный характер, подробную информацию читателю придется поискать где-то в другом месте. Данное эссе — от начала до конца — одно сплошное предисловие.

*Б. Мандельброт
«Фрактальная геометрия
природы»*

Современный прогресс в реконструкции моделей нелинейной динамики из наблюдаемых временных рядов опирается главным образом на идею точечного источника сигнала. Иными словами, мы полагаем, что неизвестный нам оригинал, который задан в виде некоторой группы преобразований в многомерном фазовом пространстве, непрерывно продуцирует одномерную «тень» или «наблюдаемую» в форме временного сигнала или реализации, т. е. потока событий или меток на оси времени. Поскольку непрерывность сохраняет отношение близости, образы предельных режимов системы, аттракторы, должны наследоваться в масштабных свойствах (скейлинге) отсчетов, меняющихся во времени. Но для траекторий на аттракторе время не играет никакой роли: все что могло произойти, уже случилось. Поэтому, наблюдаемый временной ряд это информационный код «геометрии» аттрактора. Можно попытаться, следовательно, извлечь эту геометрию перестройкой одномерных рядов в многомерные структуры. Образ n -мерной точки легко получить с помощью набора из n отсчетов временного ряда. Множество подобных образов следует теперь связать друг с другом, чтобы получить копию траектории. Оказывается, что для этого можно использовать практически любую непрерывную функцию. Простейшая из них это просто сдвиг. Вот в принципе и весь рецепт получения универсальной модели «à la Takens». Полученная конструкция «по образу и подобию» эквивалентна в четко определенном смысле оригиналу — аттрактору, и, следовательно, ее можно предъявить в качестве «гладкого» объяснения наблюдаемой.

Однако, в большинстве случаев мы имеем дело с «распределенными» динамическими системами. Они демонстрируют не только сложное поведение во времени, но имеют нетривиальную пространственную структуру. Другими словами, проекциями их динамики в «Мир Экспериментатора» являются не только (и не столько) одномерные временные ряды, но и «мгновенные снимки — snapshots», изображения или «сцены». Таким образом приходится иметь дело уже с двумя видами сложности: временной, которая отслеживается временными рядами каких-либо интегральных параметров, и пространственной, которая «кодируется» геометрией и топологией «сцен». Последние также меняются во времени, так что обычно имеют дело с временной последовательностью «мгновенных снимков», где термин «мгновенный» зависит конечно от контекста эксперимента. Соответствующие массивы данных могут иметь произвольную форму: прямые фотографии или видеозаписи, цифровые изображения или просто числовые матрицы, строки которых содержат скалярные или векторные значения измеряемого поля в пространственно различных точках процесса или системы. Собирательным синонимом такого разнообразия является понятие «паттерна». Подходящее определение этого термина я нашел в замечательной научно-популярной книге *Грея Уолтера* [1] и приведу его частично: «... понятие «паттерн» подразумевает любую последовательность явлений во времени или любое расположение предметов в пространстве, которые можно отличить от другой последовательности или другого расположения или сравнить с ними. Первое существенное свойство паттерна заключается в том, что вы можете вспомнить его и сравнить с другим паттерном. Это свойство отличает паттерн от беспорядочных явлений или хаоса».

В этой лекции пойдет речь только о пространственных паттернах. Точнее, я намереваюсь рассказать о некоторых современных математических инструментах для «арифметизации» и анализа таких объектов. Следует пояснить, чем же плохи уже знакомые подходы, которые (или под которые) можно подогнать распределенные системы. Проще всего «подогнать» объект, попытаться, например, загнать распределенную систему в прокрустово ложе точечного источника. Для этого достаточно «свернуть» паттерн так, чтобы геометрия выражалась «интегралом» по всему изображению. Именно такой подход был реализован, например, в наблюдательной программе «Солнце как звезда». Предположим, что нас интересует изменение яркости фотосферы или напряженность магнитного поля Солнца. Их измеренные значения зависят от положения на видимой половине сферы. Вся эта информация доступна, но ее трудно осмыслить: слишком велико

многообразие деталей по сравнению с наблюдениями звезд. Идея, которая позволила упростить ситуацию, оказалась настолько простой, что за ней нужно было только нагнуться. Перед телескопом был установлен небольшой металлический шарик с зеркальной поверхностью. Отраженное от него изображение Солнца, которое попадало в объектив было практически точечным! Понятно, что при таком «интегрировании» вся информация о геометрической сложности теряется, однако полученные временные ряды яркости или интегрального магнитного потока можно сравнивать с аналогичными для звезд. Другой путь — адаптировать не сам объект, а методы; например, обобщить «скалярную» эмбедологию на матричные временные ряды. Такие попытки были и оказались вполне удачными [2, 3]. Используя обобщение алгоритма Такенса на двумерную наблюдаемую (скалярное поле), для последовательности мгновенных снимков системы градиентного типа удалось, в частности, показать [3], что целочисленная корреляционная размерность снимка соответствует «эволюционно зрелому» движению жидкости, а дробная размерность — переходным, турбулентным режимам. Правда, на этом пути много вычислительных трудностей, да и статистическая достоверность полученных оценок далека от приемлемого стандарта, достигнутого для скалярных рядов. Существует и третий подход. Он заключается в том, чтобы корректным образом извлечь полезную динамическую информацию из геометрии и топологии пространственных паттернов, а затем представить ее в форме привычных временных рядов. Необходимые для этого методы можно найти в Вычислительной, Интегральной и Стохастической Геометриях, которые совсем недавно пополнились инструментами Вычислительной Топологии.

Было бы наивно полагать, что одна лекция может дать последовательное введение во все упомянутые предметы. Все, что могло получиться в лучшем случае — это глоссарий с картинками к каждой из упомянутых областей, написанный с точностью «до технических деталей».

Автор искренне благодарен *Ольге Круглун и Екатерине Данилкиной* за помощь в техническом оформлении рукописи и редактору Лекций *Ю. В. Тюменцеву* за многочисленные исправления.

Вычислительная геометрия

Назовем первую часть это, вторую часть то, а переход от одной к другой назовем не то и не это. Назовем не то и не это «препятствием».

Д. Хармс

«О явлениях и существованиях»

Вычислительная геометрия¹⁽¹⁾ имеет дело с дизайном и вычислительной сложностью алгоритмов для решения геометрических задач при различных моделях вычислений. Я приведу простой пример, в котором одна и та же задача имеет две таких модели. Рассмотрим точку $P(x_p, y_p)$ и линию $l: y = ax + b$ на плоскости (рис. 1). Необходимо определить, выше или

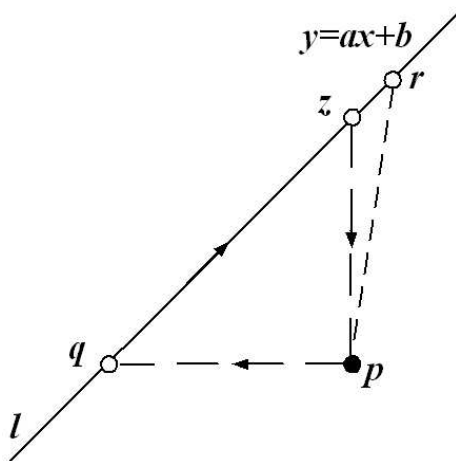


Рис. 1. Две модели вычисления

ниже прямой l лежит точка P .

¹Если ссылка на примечание представляет собой число, заключенное в круглые скобки, например, (3), то она обозначает номер примечания, помещенного в конце данного текущего раздела. Ссылка в виде числа, не заключенного в скобки — обычное подстраничное примечание. — Прим. ред.

Прежде всего предположим, что l находится в «общем положении», т. е. не вертикальна и не горизонтальна. Тогда достаточно найти точку пересечения (x_z, y_z) вертикали, проходящей через P , с прямой l ; ее координаты удовлетворяют уравнениям: $y_z = ax_p + b$; $x_z = x_p$. Очевидно, что P лежит ниже l , если $y_z > y_p$, и выше ее, если $y_z < y_p$. Но это лишь один подход к решению. Другой заключается в следующем: отметим на линии две дополнительные точки — q и r . Тогда три точки p, q, r образуют треугольник. Предположим, что мы располагаем программой для вычисления площади S треугольника по «школьной» формуле:

$$S = \frac{1}{2} [x_q(y_r - y_p) + x_r(y_p - y_q) + x_p(y_q - y_r)].$$

В случае, если $S = 0$ — точка P лежит на l . Заметим теперь, что порядок индексов точек q, r и p в слагаемых нашей формулы соответствует обходу треугольника по часовой стрелке, как показано на рисунке, и приводит к отрицательному знаку площади. Однако, если P лежит выше линии, индексы вершин треугольника соответствуют противоположной ориентации и площадь будет положительна. Таким образом, для ответа нужна даже не сама площадь, а только ее знак. Итак, мы имеем два разных подхода к решению. Методы вычислительной геометрии позволяют оценить, какой из них работает быстрее, требует меньшего объема памяти и более устойчив к ошибкам вычислений.

Теперь обратимся к самим алгоритмам. Нас интересуют прежде всего те из них, которые позволяют наделить пространственный паттерн некоторым «скелетом», позволяющим, по меньшей мере, проделать таксономический анализ, т. е. разделить наблюдаемое множество на классы и присвоить классам названия². Начиная с *Карла Линнея*⁽²⁾, считается, что образ эквивалентен набору правил для его распознавания. В основу таких правил проще всего положить алгоритмы «близости», геометрической или топологической. Их традиционный набор из вычислительной геометрии включает диаграмму Вороного (ДВ), триангуляцию Делоне (ТД) и минимальное остовное дерево (ЕМОД). Современным дополнением к ним являются альфа-формы (АФ). Рассмотрим эти алгоритмы в упомянутом порядке.

²Любопытно, что еще у римлян была поговорка *Nomen est numen* — «назвать — значит узнать».

Диаграмма Вороного

Представим себе город, на территории которого случайным образом расположено конечное множество $P \in R^2$ обменных пунктов валюты. Разумно полагать, что жители посещают ближайший к ним пункт. В этом случае, вся территория естественным образом разбивается на *локусы* V_p — окрестности каждой из точек P такие, что расстояние любой точки локуса $x \in V_p$ до его центра p меньше, чем расстояние до любого другого центра $q \in P$. Формально,

$$V_p = \{x \in R^2 | \forall q \in P - \{p\}, \|x - p\| \leq \|x - q\|\},$$

где $\{p\}$ означает, что точка p удалена из P . В общем случае эта задача была решена в 1908 году русским математиком *Георгием Вороным*⁽³⁾, поэтому искомое разбиение принято называть *диаграммой Вороного*³. Если P содержит всего две точки, скажем p и q , то диаграмма тривиальна: надо просто провести перпендикуляр через середину отрезка, соединяющего эту пару. Тогда локусами будут две полуплоскости, содержащие p и q соответственно. В общем случае диаграмма Вороного (ДВ) образуется множеством пересечения таких полуплоскостей. Обычно, алгоритмы для построения ДВ предполагают, что точки находятся в общем положении; в данном случае это означает, что никакие четыре точки множества P не лежат на одной окружности. Легко доказать, что в каждой вершине ДВ пересекается в точности три ребра диаграммы. Более того, любая из вершин ДВ является центром окружности, на которой лежат ближайшие к ней три центра локусов; при этом внутри окружности нет ни одного центра. Существует один быстрый алгоритм для построения ДВ. Его оригинальность заключается в том, что он скорее «аналоговый», чем цифровой.

Метод степного пожара. Идея алгоритма берет начало из практики тушения степных пожаров. Известно, что наиболее эффективный способ остановить огненный фронт — поджечь степь по его другую сторону так, чтобы две огненных стены пришли в соприкосновение и погасили друг друга. Предположим теперь, что мы подожгли одновременно все точки конечного множества P . Горящий фронт распространяется из каждой точки в виде расширяющегося со временем диска. Мы полагаем при этом, что «степь» однородна, так что увеличение радиусов дисков происходит синхронно и с постоянной скоростью так, что два соседних круговых фронта

³Синонимами являются: ячейки Дирихле, многоугольники Тиссена, ячейки Вигнера-Зейтца.

встретятся точно посередине интервала, соединяющего их центры. Сперва пламя погаснет в точке контакта; остальные точки фронтов будут продолжать движение до встречи со своим антиподом. В результате последовательность точек гашения образует прямую линию, которая делит пространство между точками на две равные полуплоскости, и множество таких линий образует то, что нам нужно, т. е. ДВ. В качестве примера на рис. 2 показаны три последовательных кадра степного пожара для множества P , которое содержит три точки.

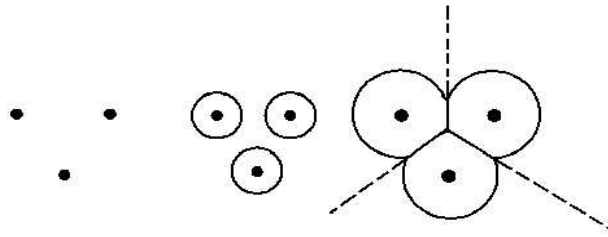


Рис. 2. Степной пожар для трех точек

Различные варианты для геометрии фронтов, а следовательно, разных разбиений, можно получить, рассматривая горение в L_p -метриках, при $p \neq 2$. Например, метрика L_∞ приведет к разбиению плоскости на множество неравных прямоугольников⁽⁴⁾.

Триангуляция Делоне

Следующий алгоритм близости, тесно связанный с ДВ, впервые появился в практической картографии. Считается, что триангуляцию впервые применил в 1915 году нидерландский астроном и математик *Виллеброрд Снеллиус*⁽⁵⁾ при прокладке каналов для градусных измерений. Математические основы алгоритма были созданы советским математиком *Борисом Делоне*⁽⁶⁾. Соединим отрезками все пары центров ячеек Вороного (рис. 3). Полученный граф, двойственный⁴ диаграмме Вороного, и называют *триангуляцией Делоне (ТД)*. В общем случае любая триангуляция — это разбиение компактной области на плоскости или на многообразии на систему

⁴Чаще говорят — дуальный.

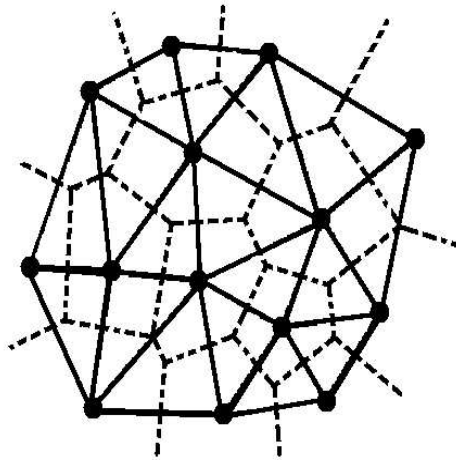


Рис. 3. Триангуляция Делоне вместе с диаграммой Вороного

простых треугольных ячеек-симплексов⁽⁷⁾. Однако, не всякая триангуляция является ТД: последняя использует «наиболее правильные» треугольники в следующем смысле. Рассмотрим два смежных по ребру треугольника, входящих в ТД и образующих четырехугольник. Исходное ребро можно перебросить⁵ так, чтобы оно соединило две других вершины. Так вот в ТД минимальный из шести углов в паре треугольников при этом не увеличится! Кроме того, оказывается, что для того, чтобы триангуляция была ТД необходимо и достаточно, чтобы внутри окружности, описанной вокруг любого из треугольников, не было ни одной другой вершины ТД — это свойство называют «круговым критерием» ТД.

Триангуляция Делоне чрезвычайно полезна для задач интерполяции. Предположим, что нам известны значения некоторой гладкой функции $f(x, y)$ на конечном множестве опорных точек: $f(x_i, y_i) = f_i$, $i = 1, 2, \dots, N$. Пусть эти измеренные значения функции получены на произвольной, не обязательно регулярной, сетке аргументов. Очевидно, что задача восстановления $f(x, y)$ по конечному обучающему множеству не является корректной. В самом деле, к любой найденной поверхности, проходящей через опорные точки, можно добавить член $\varphi(x, y) \prod_{i=1}^N (x - x_i)(y - y_i)$,

⁵Это называется — «сделать флип».

где $\varphi(x, y)$ — произвольная функция. Для уменьшения неопределенности, вызванной осцилляциями функции между опорными точками, сетку измерений желательно уплотнить и сделать ее по возможности эквидистантной. Для этого построим ТД по опорным точкам. Теперь добавим в каждый треугольник по одной новой точке. Соединим каждую такую точку с вершинами треугольника, который является ее окрестностью. Используя круговой критерий, перестроим полученную измельченную триангуляцию так, чтобы она тоже стала ТД. После этого, вычислим значения нашей функции в новых точках, используя какие-либо робастные методы интерполяции. Если это необходимо, повторим процедуру измельчения. В результате мы будем иметь более представительное обучающее множество, заданное на сетке аргументов, близкой к эквидистантной. Теперь для восстановления $f(x, y)$ можно с успехом использовать, например, искусственную нейронную сеть.

Евклидово минимальное остовное дерево (ЕМОД)

Формально ЕМОД получается при решении следующей задачи. На плоскости заданы N точек. Необходимо построить дерево⁽⁸⁾, вершинами которого являются все эти точки при условии, что суммарная длина всех ребер графа минимальна. Практически такое дерево может быть реализовано, например, телефонной сетью минимальной длины, которая соединяет N абонентов. Пример ЕМОД приведен на рис. 4. Прямой, но не оптимальный

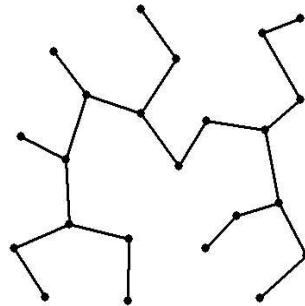


Рис. 4. Минимальное остовное дерево для 23 точек

путь нахождения ЕМОД заключается в следующем. Выбираем одну из N

точек, скажем p , и вычисляем все расстояния до $N - \{p\}$ точек. Переходим в ближайшую к p точку, т. е. проводим ребро, и повторяем вычисление матрицы расстояний. Начиная с третьего шага, при выборе ближайшего соседа проверяем отсутствие цикла. Продолжаем упомянутые операции пока не получим дерево, содержащее все точки множества. Лучший из известных алгоритмов ЕМОД имеет сложность $O(N^2)$. Практически очевидно, что ЕМОД является подграфом ДВ, которая задает отношения близости между точками.

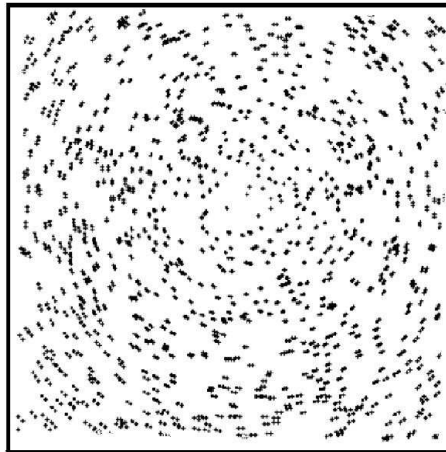


Рис. 5. Glass pattern

ЕМОД — наиболее излюбленный граф для приложений, диапазон которых простирается от задач кластеризации до оценок размерностей Реньи в мультифрактальных множествах и вычисления индекса несвязности. Наиболее интересные факты, связанные с этим графом, обусловлены тем обстоятельством, что наш мозг, по-видимому, реализует аналогичный алгоритм: неупорядоченный набор близких пятен или точек мы видим как фрагменты линейных структур. Рис. 5 демонстрирует этот замечательный эффект⁶ на примере наложения двух идентичных панелей, содержащих случайное распределение точек, которые наложены друг на друга после поворота одной из них на небольшой угол. Концентрические круги, которые вы видите —

⁶Его называют Glass pattern.

это следствие реализации алгоритма ЕМОД нашим зрением.

В астрономии наиболее известным проявлением Glass pattern'а были знаменитые каналы Марса. Скопление темных точек на диске планеты наблюдатели на пределе разрешения видели как сетку линий или каналов. На рис. 6 приведена карта таких каналов, построенная французским астрономом Эженом Антониади⁽⁹⁾ по наблюдениям Марса в эпоху великого противостояния 1909 года. Другим менее известным проявлением ЕМОД

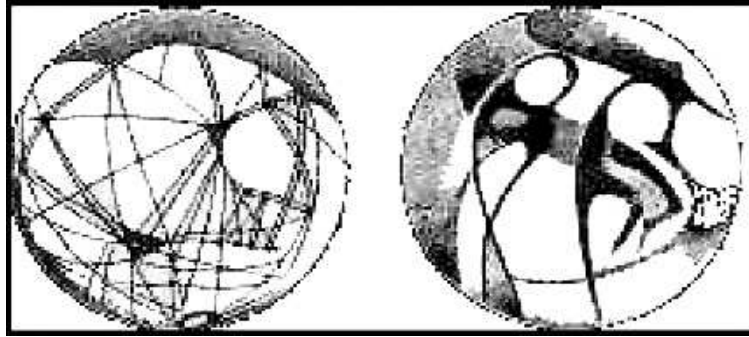


Рис. 6. Карта марсианских каналов по Антониади

является «инвариантная» форма созвездий на картах звездного неба у разных народов. Компьютерные эксперименты показали, что если задать установленные границы созвездий и построить ЕМОД для ярких звезд в этой области, мы получим стандартные формы созвездий с точностью до различных мифологических интерпретаций. ЕМОД для семи наиболее ярких звезд Большой Медведицы, например, имеют форму ковша.

Альфа-формы

Обратимся теперь к алгоритму, который позволяет «приписать форму» дискретному множеству. Понятно, что никакой формы такое множество не имеет, и иметь не может. Однако, в ряде задач такая форма крайне желательна. Например, в биофизике каждый атом молекулы белка можно представить как твердую сферу. В этом случае молекула будет изображаться как 3D конфигурация шаров разного радиуса. При описании взаимодействия между молекулами соответствующие уравнения должны содержать такие

величины, как поверхность и объем молекулы. Совершенно непонятно, как получить их из «скелета» молекулы. Герберт Эдельсбрунер предложил в 90-х годах алгоритм близости, который позволяет получить непрерывную последовательность форм, параметризованную некоторым числом α . Эти формы моделируют геометрию и топологию произвольной точечной конфигурации с желаемой точностью и, конечно, имеют площади и объемы.

Идею алгоритма можно понять из следующего примера. Представим себе однородно закрашенную доску, в которую забиты гвозди. Каждый гвоздь — это точка нашего множества и конфигурация гвоздей случайна. Выберем круглую резинку с радиусом α . Используя ее, сотрем краску на нашей доске там, где гвозди не препятствуют движению резинки. Если $\alpha = 0$, мы конечно получим исходную конфигурацию. Если резинка превышает максимальный размер между гвоздями, это произойдет, например, при $\alpha = \infty$ — мы получим выпуклую оболочку конфигурации. Это и будут две предельные α -формы (АФ). Все промежуточные АФ получаются при изменении α в диапазоне $0 < \alpha < \infty$. На рис. 7 приведен пример шести α -форм, построенных на конфигурации точек, образующих греческую букву α . АФ тесно связаны с триангуляцией Делоне. ТД позволяет

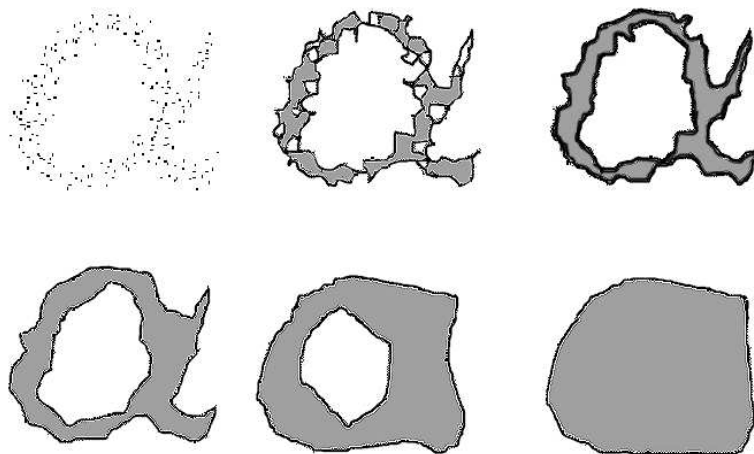


Рис. 7. Последовательность α -форм для конфигурации буквы α

дополнительно ввести еще одну полезную топологическую структуру — α -комплекс, который позволяет вычислять даже такие тонкие вещи, как

инварианты алгебраической топологии, например, числа Бетти. Алгоритм построения α -комплекса заключается в следующем. Для точечной конфигурации сперва получают ТД. Затем с помощью диска радиуса α проверяется каждая из сторон треугольника. В α -комплекс включается та часть

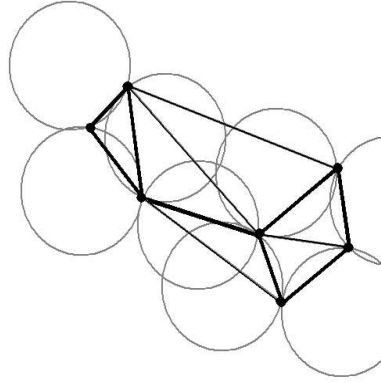


Рис. 8. α -комплекс для множества, содержащего 7 точек

триангуляции Делоне (ее отдельные ребра и грани), которая называется « α -незащищенной»⁷. Другими словами, ребро триангуляции принадлежит α -комплексу, если диски радиуса α , соответствующие ребру, являются пустыми, т. е. не содержат ни одной вершины ТД. На рис. 8 приведен пример α -комплекса, построенного на семи точках.

Нейронный газ и топология близости

Используя правило обучения Хебба, дополненное конкуренцией, можно построить нейронные сети (НС), позволяющие получать разбиение Вороного и триангуляцию Делоне. В этой лекции мы вынуждены ограничиться только краткими замечаниями; подробности можно найти в работах [26, 27]. Рассмотрим НС с элементами $i, j = 1, \dots, N$, снабженными латеральными связями, величина которых задается матрицей C_{ij} . Если $C_{ij} > 0$, считается, что нейрон i связан с нейроном j . Правило обучения Хебба имеет вид $\Delta C_{ij} \propto y_i y_j$, где y_i, y_j — входная и выходная активность

⁷ α -exposed.

нейрона соответственно. Для входного вектора \mathbf{v} с распределением $P(\mathbf{v})$ полагают, что $y_i = R(\|\mathbf{v} - \mathbf{w}_i\|)$, где R — стандартная функция активации, а \mathbf{w} — вес нейрона. Правило Хебба модифицируется следующим образом: $\Delta C_{ij} \propto y_i y_j$, если $y_i y_j \geq y_k y_l$ и $\Delta C_{ij} = 0$ в противном случае. Такое правило называют *конкурирующим* правилом Хебба. Можно показать [26, 27], что структура связей, соответствующая этому правилу, будет совпадать с ТД на множестве весов $\{\mathbf{w}\}$. Точнее, элементы C_{ij} в процессе обучения стремятся к асимптотике $\theta[C_{ij}(t \rightarrow \infty)] = A_{ij}$, где θ — функция Хевисайда, а A_{ij} является матрицей инцидентности для ТД на множестве весов $\{\mathbf{w}\}$ так, что $A_{ij} = 1$, если $V_i \cap V_j \neq \emptyset$, где V_i, V_j — ячейки Вороного для весов w_i, w_j . Существуют компьютерные пакеты, реализующие описанный подход. Наиболее популярными НС для получения ТД и ДВ являются НС типа «нейронный газ», введенный Б. Фритцке [28] и его модификация, известный как «растущий нейронный газ». В качестве примера на рис. 9 приведена диаграмма Вороного для кольца, полученная с помощью растущего нейронного газа.

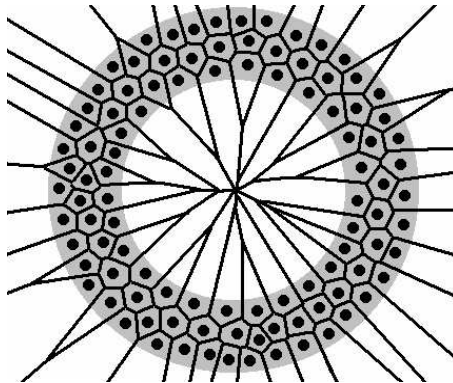


Рис. 9. ДВ кольца, полученная с помощью «растущего нейронного газа»

Примечания

1. Геометрические алгоритмы реализуются на компьютере, поэтому, следуя Е. Дейкстре *компьютерная наука должна называться вычислительной наукой по тем же причинам, по которым хирургию не называют наукой о ноже*. Считается, что современная вычислительная геометрия началась в 1978 году с докторской диссертации

М. Шеймоса, которая легла в основу классического учебника [5]. Однако, еще в 1969 году *М. Минский* и *С. Пейперт* исследовали некоторые геометрические алгоритмы в контексте распознавания образов перцептронами [43]. На практике геометрические алгоритмы использовались уже в глубокой древности. Так, египетские архитекторы еще 4000 лет назад использовали своеобразный «компьютер» для построения прямого угла на местности. Он представлял собой веревку со связанными концами, разделенную двенадцатью узлами на равные промежутки. Прямой угол строился в соответствии с теоремой Пифагора $(4)^2 + (3)^2 = (5)^2$ при одном фиксированном катете.

2. *Линней Карл* (1707–1778 гг.), шведский естествоиспытатель, первый президент шведской АН. Автор ряда трудов по ботанике, зоологии, медицине. Получил мировую известность благодаря созданной им системе классификации растительного и животного мира. Она была описана в “*Systema naturae*” и основана на бинарной номенклатуре, в которой каждый вид обозначается двумя латинскими названиями — родовым и видовым. Линней открыл и описал около 1500 растений, разделив их на 24 класса. Всех животных Линней поместил в 6 классов: **Млекопитающие, Птицы, Амфибии, Рыбы, Черви, Насекомые**. Человек был отнесен к классу млекопитающих, к отряду приматов. Для сравнения, в книге *У. Гренандера* [16] можно найти любопытный пример старинной китайской таксономии животных. Они делятся на «принадлежащих императору, сказочных, неисчислимых, нарисованных очень тонкой кисточкой из верблюжьей шерсти, буйствующих, как в безумии и т. д.» Сирены, бродячие собаки и молочные поросята образуют отдельные классы.
3. *Вороной Георгий Феодосьевич* (1868–1908). Русский математик, член-корреспондент Петербургской АН (1907). В 1889 году, по окончании Петербургского университета был оставлен для подготовки к профессорскому званию. В 1894 году защитил магистерскую диссертацию «О целых алгебраических числах, зависящих от корня неприводимого уравнения 3-й степени» и был назначен профессором университета и политехнического института в Варшаве. В 1897 году защитил докторскую диссертацию «Об одном обобщении алгоритма непрерывных дробей», удостоенную премии имени Буняковского. Сделал ряд важных открытий в геометрии многогранников и геометрии чисел.

4. Алгоритм степного пожара использовался для кодирования графических символов и контурных изображений [8]. Представим себе идеально однородную лужайку, в форме квадрата A , сухую и готовую загореться. За пределами периметра травы нет. Если поджечь периметр одновременно во всех точках, огонь будет гаснуть на диагоналях квадрата и мы получим «скелет» A в форме буквы \times . Сжигание прямоугольной лужайки приведет к дереву $\succ<$. Аналогичным образом при сгорании произвольного контура мы получим скелеты в форме планарных графов или деревьев, которые легче представить в памяти ЭВМ, чем оригинал.
5. *Виллеброрд Снеллиус (Снелль)* (латинизированное *Snellius* — нидерландское *ван Снел ван Ройен*) (1580–1626), нидерландский астроном и математик. Предложил метод триангуляции для определения расстояния по пересеченной местности и впервые применил его для определения длины дуги меридиана, используя треугольники с расстоянием до вершин в 30–40 км. В 1614–1617 годах произвел измерение длины дуги в 1 градус в Голландии, получив величину в 107335 м. В 1621 году экспериментально открыл закон преломления света. В 1624 году ввел «Локсодромию» — кривую, пересекающую глобус (сферу) под постоянным углом.
6. *Делоне Борис Николаевич* (1890–1980). Родился в Петербурге. Советский математик, член-корреспондент АН СССР (1929), член Немецкой академии исследователей природы «Леопольдина» (1962). Окончил Киевский университет (1913); в 1922–1935 годах — профессор Ленинградского, в 1935–1942 годах — Московского университетов. Основные работы по теории чисел, по алгебре — геометризации теории Галуа, по геометрии — теории правильного разбиения пространства.
7. Пусть e^2 — евклидов 2-симплекс (треугольник) и $\Delta = \{e_i^2\}$ — множество таких треугольников, определенных в некоторой области $M \subset R^2$. Множество Δ называют *триангуляцией* M , если выполняются следующие условия:
 - (а) Для каждой точки $P \in M$ найдется по крайней мере один e_i^2 , содержащий P .

- (b) Если $P \in e_i^2$, но не лежит на его стороне, то e_i^2 — единственный треугольник, содержащий P .
- (c) Если P принадлежит стороне e^1 треугольника $e_1^2 \in \Delta$, но P не вершина e_1^2 , то существует в точности еще один треугольник $e_2^2 \in \Delta$ такой, что $e_1^2 \cap e_2^2 = e^1$.
- (d) Если P — вершина e_1^2 , то существует конечное число треугольников $e_1^2, e_2^2, \dots, e_n^2$, имеющих P своей вершиной. Эти треугольники образуют «звездную» окрестность точки P .

8. *Деревом* называется связный граф, не содержащий циклов; такой граф, в частности, не имеет и кратных ребер. Для каждой пары вершин дерева существует единственная соединяющая их цепь. Если граф, вообще говоря, несвязный и не содержит циклов, то каждая его связная компонента будет деревом; пользуясь терминологией, принятой в ботанике, такой граф называют *лесом*.

9. *Антониади Эжен Мишель* (1870–1944), французский астроном. Родился в Константинополе, где с 18 лет самостоятельно занимался астрономией. В 1893 году Антониади приехал в Париж и стал ближайшим сотрудником *К. Фламариона* в его частной обсерватории, где проводил наблюдения Марса, который привлек всеобщее внимание, после того как в 1870-е годы *Скиапарелли* обнаружил на Марсе «каналы». В 1902 году Антониади был приглашен руководить секцией Марса в Британской астрономической ассоциации. Сделал замечательно точные зарисовки поверхности планеты. В 1904 году Антониади занимался археологическими раскопками в Софийском соборе Константинополя и провел ряд исследований по истории астрономии в Древней Греции и Египте. Награжден рыцарским крестом Почетного легиона. В 1928 году он принял французское гражданство. Имя Антониади занесено на карты Луны и Марса.

Путеводитель по литературе. Классическим учебником по вычислительной геометрии на русском языке является [5]. Популярное введение можно найти на веб-странице [6]; более содержательное — в обзоре *Р. Клетте* [7]. Метод степного пожара для получения скелетов описан в статье [8]. Применение ТД для построения карты Фобоса и история пакета «ГЕОФОРМ» содержатся в статье [9]. Топологические определения триангуляции можно найти, например, в классическом учебнике по топологии [10]. В статьях

[11–12] приводятся примеры использования диаграммы Вороного в космологии. Использование алгоритма ЕМОД для кластерного анализа содержится в статьях [13–14]. В работе [15] описано применение этого алгоритма для нахождения линейных структур (filamentary structure) в крупномасштабном распределении галактик. Абстрактному изложению математической таксономии посвящена монография У. Гренандера [16]. Литература по α -формам на русском языке отсутствует. Хорошим введением в этот предмет служат статьи [17–21] и диссертация Ванессы Робинс [22]. α -формы в 3D описаны в статье [23], а их приложения к биологии [24] и триангуляции [25] можно найти на веб-страничке Г. Эдельсбрунера. Применение нейронного газа к топологии описано в статьях [26, 27]. Демонстрационный пакет с сопутствующими публикациями доступен на сайте [28].

Геометрические вероятности: Игла Бюффона и парадокс Бертрана

К понятию вероятности мы прибегаем, когда не знаем чего-либо с полной уверенностью. Но неуверенность эта носит либо субъективный характер (я не знаю, что произойдет, но кто-то другой возможно и знает), либо объективный (никто не знает и знать не может). Субъективная вероятность — это протез при информационном увечье.

Ст. Лем

«О невозможности жизни»

Знаменитый французский ученый, автор «Естественной истории» Жорж Луи де Бюффон⁽¹⁾ рассмотрел в 1733 году задачу, которая стала предтечей современных методов Монте Карло. На плоскость нанесена сетка параллельных прямых, разделенных расстоянием $2a$. Игла длины $2l < 2a$ брошена случайно⁸ на плоскость. Какова вероятность того, что игла пересечет одну из прямых? Соображения симметрии позволяют ограничить-

⁸Т.е. так, что результат не зависит от наших намерений.

ся лишь одним фрагментом с двумя прямыми (рис. 10). Положение иглы определяется углом θ между ее направлением и расстоянием до ближайшей прямой. Центр иглы P может занять произвольное положение между прямыми с одинаковой вероятностью, а сдвиг иглы по вертикали не влияет на пересечение. При фиксированном значении угла θ вероятность пересечения равна $2x/2a$. Можно считать, что угол θ равномерно распределен в интервале $[0, \pi/2]$, так как пересечение сохраняется и при угле $\pi - \theta$. Следовательно, надо найти среднее значение величины $x/a = (l/a) \cos \theta$, т. е. математическое ожидание:

$$\frac{l/a}{\pi/2} \int_0^{\pi/2} \cos \theta d\theta = 2(2l)/2a\pi.$$

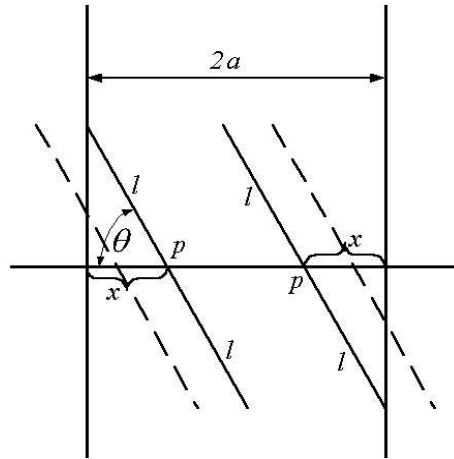


Рис. 10. Случайная игла на решетке

Таким образом, получаем, что вероятность P того, что игла пересечет прямую, равна:

$$P = \frac{2(\text{длина иглы})}{\text{длина окружности радиуса } a}.$$

Популярность этой задачи исторически связана с возможностью экспериментального определения⁽²⁾ числа π . Для нас важно другое: искомая

вероятность выражается отношением двух геометрических мер, так что случайными величинами являются не числа, как в комбинаторной теории вероятностей, а геометрические объекты. Оказывается, что в этом случае очень важно корректным образом определить понятие «случайный». Следующий пример, принадлежащий Жозефу Бертрону⁽³⁾, показывает, что нечеткость в понимании случайного множества геометрических элементов приводит к парадоксам.

Задача Бертрана состоит в нахождении вероятности того, что длина «случайной хорды» в окружности единичного радиуса превзойдет сторону вписанного равностороннего треугольника. Рассмотрим три способа выбора случайной хорды (см. рис. 11).

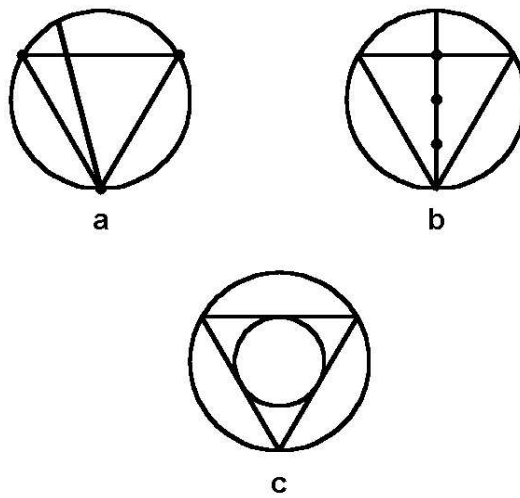


Рис. 11. Три способа выбора случайной хорды

Первый способ. Исходя из соображений симметрии, можно считать, что одним концом хорды является произвольная фиксированная точка окружности (рис. 11a), например, вершина вписанного треугольника. Выберем другой конец случайно с равномерным распределением по окружности. Вершины треугольника делят окружность на три равные дуги и случайная хорда будет длиннее стороны правильного треугольника, если она пересекает этот треугольник. Так что искомая вероятность будет равна $1/3$.

Второй способ. Зафиксируем диаметр, к которому перпендикулярна хорда. Будем считать фиксированным именно этот угол наклона хорды к диаметру и исследуем только одно семейство параллельных хорд. Мы хотим знать, какая часть этого семейства хорд расположена удачно с нашей точки зрения (рис. 11b), т. е. хорды окажутся длиннее чем $\sqrt{3}$. Элементарное исследование вписанного в круг равностороннего треугольника показывает, что если разделить диаметр на четыре части, то хорды, пересекающие его в двух крайних четвертях, не отвечают необходимым требованиям, тогда как хорды, пересекающие центральные две четверти, — «удачны». Таким образом, из четырех равновероятных сечений диаметра два «удачны», а два нет. Это означает, что вероятность «успеха» равна $1/2$.

Третий способ. Случайным образом (равномерно) в круге выберем точку, которая определяет единственную хорду, ту, что является ее серединой. Эта хорда длиннее стороны вписанного правильного треугольника тогда и только тогда, когда ее середина лежит внутри круга, вписанного в треугольник (рис. 11c). Радиус этого круга равен $1/2$; следовательно, мера возможных положений центра хорды — площадь вписанного круга, которая составляет $1/4$ площади исходного. Таким образом, вероятность того, что случайно выбранная точка определяет «удачную» хорду, равна $1/4$.

Итак, мы получили разные ответы: $1/3$, $1/2$ и $1/4$, каждый из которых справедлив в рамках своего понятия «случайности». В теории вероятности случайный выбор понимают в контексте заданной плотности вероятности. Следовательно, нам необходимо нечто аналогичное для геометрических элементов — точек, прямых, плоскостей, и т. д. Такие аналоги называют *геометрическими мерами* и изучаются в интегральной геометрии.

Примечания

1. *Жорж Луи Леклерк, граф де Бюффон* (1707–1788), французский естествоиспытатель, интендант королевского ботанического сада, академик (1753), автор «Истории и теории Земли». Известен как выдающийся стилист-теоретик; фраза из его речи при вступлении во французскую академию «стиль — это человек», стала крылатой [32]. Вот, например, как Бюффон описывал попугая: «Болтать не значит говорить; слова становятся языком, лишь если выражают и сообщают другим мысль говорящего. Попугай же, хотя и трещат с беспримерной легкостью, не наделены умом, выражением коего в слове одно только и создает высокую способность говорения». Задачу об игле

Бюффон рассмотрел в 1733 году в труде «Опыт моральной арифметики», который был опубликован лишь в 1777 году.

2. Оценка основана на выражении $\pi = 2l/\hat{P}a$, где $\hat{P} = N_1/N$ — несмещенная оценка для P , полученная при N_1 пересечениях из N испытаний. Результаты множества таких экспериментов резюмируются в работе Гриджемана [33]. Лучший результат, который совпал с известным в Древнем Китае приближением, дал $\pi \approx 355/113 \approx 3.1415929$, при $N = 3408$ и явился большой удачей, обусловленной выбором подходящего момента остановки испытаний. М. Кендалл [29] заметил, что, если точное значение π было бы неизвестно, так что остановка испытаний в благоприятный момент не могла быть использована, то оценка его с помощью измерения лентой деревянного круга дала бы более надежные результаты.
3. Бертран Жозеф Луи Франсуа (1822–1900), французский математик, иностранный почетный член (1896) Петербургской АН.

Путеводитель по литературе. Хорошее введение в геометрические вероятности можно найти в монографии [29], где рассмотрена и задача Бюффона об игле. Парадокс Бертранна рассмотрен в книге [30] и статье [31]. Речь Бюффона в Академии, в русском варианте, доступна на веб-странице [32].

Интегральная геометрия: Случайные точки и случайные прямые в R^2

Любое мировоззрение строится на
какой-нибудь дюжине координат и
констант из множества нам
неизвестных.

Л. Леонов
«Мироздание по Дымкову»

Геометрические меры, так же как и любые другие меры, должны удовлетворять некоторым разумным свойствам. Для того, чтобы уточнить, что под этим понимается, начнем с самого простого — случайных точек на плоскости. Припишем каждой точке некоторое малое неотрицательное число

m . Тогда сумма этих чисел для всей плоскости будет бесконечной, так что это не вероятностная мера. Но это еще не повод для беспокойства. Наложим на меру некоторые ограничения. Прежде всего, следуя классическим традициям, позаботимся об аддитивности меры. Для этого рассмотрим конечное или счетное множество непересекающихся подмножеств точек $A_1, A_2, \dots, A_n, \dots$. Аддитивность меры означает выполнения следующего условия: суммарная мера всех подмножеств равна сумме мер каждого из них, т. е. $m(\bigcup A_i) = \sum m(A_i)$.

Поскольку мы будем иметь дело с геометрическими телами, вторым разумным условием будет совпадение мер конгруэнтных множеств. Иными словами, если множество A_1 с помощью поворотов и перемещений на плоскости можно совместить с множеством A_2 , оба множества должны иметь одинаковую меру. Иными словами, мы хотим, чтобы мера была инвариантна относительно группы твердотельных движений на плоскости⁹. Покажем, как построить меру для множества точек на плоскости, которая удовлетворяет упомянутым требованиям. Ради простоты, ограничимся мерами, которые могут быть выражены двойным интегралом вида

$$m(X) = \int_X f(x, y) dx dy.$$

Здесь X — это просто класс элементарных геометрических фигур. Поскольку интеграл — аналог суммы, аддитивность меры уже гарантирована.

Пусть x, y — ортогональные декартовы координаты точки $P \in R^2$. Тогда группа \mathfrak{G} твердотельных сдвигов и вращений на плоскости определяется известными формулами

$$\left. \begin{aligned} x &= a + x^* \cos \alpha - y^* \sin \alpha, \\ y &= b + x^* \sin \alpha + y^* \cos \alpha. \end{aligned} \right\}$$

Определим функцию $f(x, y)$ так, чтобы мера $m(X)$ была инвариантна относительно группы \mathfrak{G} . Поскольку группа \mathfrak{G} транзитивна⁽¹⁾, может существовать не более чем одна инвариантная мера, с точностью до масштабной константы. Поэтому должно выполняться равенство

$$\int_X f(x, y) dx dy = \int_{X^*} f(x^*, y^*) dx^* dy^*.$$

⁹Из трех способов выбора случайной хорды только второй (рис. 11b) удовлетворяет этому условию.

С другой стороны, используя известное правило замены переменных в двойном интеграле, имеем

$$\int_X f(x, y) dx dy = \int_{X^*} f(x, y) dx^* dy^*,$$

так как якобиан перехода к новым координатам $\frac{\partial(x, y)}{\partial(x^*, y^*)} = 1$. Из двух последних формул вытекает, что

$$\int_{X^*} f(x, y) dx^* dy^* = \int_{X^*} f(x^*, y^*) dx^* dy^*.$$

Поскольку это равенство имеет место для всякого множества X^* , необходимо, чтобы выполнялось соотношение

$$f(x, y) = f(x^*, y^*).$$

Вновь используя свойство транзитивности, из последнего равенства получаем, что функция $f(x, y)$ имеет одно и то же значение во всех точках плоскости, т. е.:

$$f(x, y) = \text{const}.$$

Полагая эту постоянную равной единице¹⁰, приходим к следующему результату:

Аддитивная и инвариантная относительно группы \mathfrak{G} мера множества X точек $P(x, y) \in X$ определяется формулой

$$m(X) = \int_X dx dy \equiv \int_X dP$$

и является единственной, с точностью до постоянного множителя.

Дифференциальная форма⁽³⁾ $dP = dx dy \doteq dx \wedge dy$ называется *плотностью* множества случайных точек¹¹. Символ *внешнего* умножения⁽³⁾. $dx \wedge dy$ позволяет не заботиться о вычислении якобиана и обобщает правило замены переменной в интеграле $\int f(x) dx \rightarrow \int f(x(u)) x'(u) du$. Действительно, если ввести новые координаты

$$x = x(u, v), \quad y = y(u, v),$$

¹⁰В этом случае меру называют «кинематической».

¹¹При вычислениях, всегда берется абсолютная величина плотности, знак которой мы опускаем.

то

$$dP = dx dy = \frac{\partial(x, y)}{\partial(u, v)} du dv.$$

Если теперь, вместо *обычного произведения* дифференциалов $dx = x_u du + x_v dv$, $dy = y_u du + y_v dv$ ¹² использовать внешнее (кососимметричное) умножение: $dx \wedge dx = dy \wedge dy = 0$; $dx \wedge dy = -dy \wedge dx$, мы получим $dx \wedge dy = (x_u y_v - x_v y_u) du \wedge dv$, где якобиан перехода к новым координатам учитывается автоматически.

Полученное выражение для плотности случайных точек позволяет прийти к ряду интересных результатов [34, 35]. Соответствующие выкладки элементарны, но слишком длинны для этих лекций. Поэтому, рассмотрим еще одну геометрическую меру, для которой проще получить приложения. Эту меру называют *плотностью множества случайных прямых на плоскости*.

Прямая линия G на плоскости определяется углом φ между направлением перпендикуляра к G с фиксированным направлением ($0 \leq \varphi \leq 2\pi$) и его длиной p (расстоянием прямой от начала O ($p \geq 0$)). Координаты p, φ являются полярными координатами основания перпендикуляра, опущенного из начала O на прямую. Уравнение прямой G тогда (рис. 12) имеет вид

$$x \cos \varphi + y \sin \varphi - p = 0.$$

Будем искать меру множества прямых в координатах (p, φ) в виде интеграла

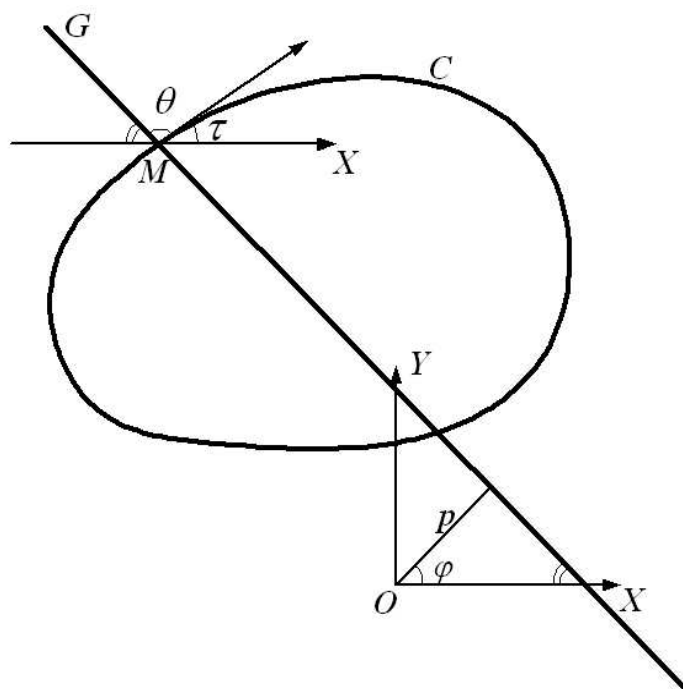
$$m(X) = \int_X f(p, \varphi) dp \wedge d\varphi.$$

Используя те же аргументы, что и для меры множества точек dP , легко получить, что плотность множества прямых определяется выражением: $dG = dp \wedge d\varphi$.

Приведем простой пример, демонстрирующий приложение этой плотности. Пусть $D \in R^2$ — компактная область с площадью F , а G — прямая, пересекающая D по хорде $G \cap D$ длиной σ . Поскольку σdp — элемент площади D , для фиксированного φ ($0 \leq \varphi \leq \pi$) легко получаем:

$$\int_{G \cap D} \sigma dG = \pi F.$$

¹²Здесь используется обозначение: $x_u = \partial x / \partial u$.

Рис. 12. Параметризация пересечения прямой G с кривой C

Пусть C — кусочно-дифференцируемая кривая на плоскости конечной длины L , которая определяется уравнениями

$$x = x(s); \quad y = y(s),$$

где параметр s есть длина дуги. Пусть G — прямая, пересекающая C в точке $M(x, y)$. Обозначим через θ угол пересечения G с кривой C (рис.12). Теперь можно выразить плотность dG через координаты s, θ , которые определяют прямую G . Из рис. 12 имеем

$$\varphi = \theta + \tau - \pi/2,$$

где τ есть угол между касательной к C и осью x .

Поскольку точка $M(x, y)$ принадлежит G , она удовлетворяет уравнению

$p = x \cos \varphi + y \sin \varphi$. Возьмем от него полный дифференциал

$$dp = \cos \varphi dx + \sin \varphi dy + (-x \sin \varphi + y \cos \varphi) d\varphi.$$

Используя очевидные соотношения $dx = \cos \tau ds$, $dy = \sin \tau ds$, получим

$$dp = \cos(\varphi - \tau) ds + (-x \sin \varphi + y \cos \varphi) d\varphi.$$

Внешнее умножение последнего выражения на $d\varphi$ дает $dG = dp \wedge d\varphi = \cos(\varphi - \tau) ds \wedge d\varphi$.

Поскольку $d\varphi = d\theta + \tau' ds$, мы приходим к равенству¹³

$$dG = |\sin \theta| ds \wedge d\theta.$$

Возьмем интеграл от обеих частей последнего равенства по множеству всех прямых, которые пересекают C . В правой части получаем

$$\int_0^L ds \int_0^\pi |\sin \theta| d\theta = 2L.$$

В левой части каждая прямая G учитывается n раз¹⁴, по числу пересечений с кривой C . Для прямых G , не пересекающих C , полагаем $n = 0$. Если кривая C не имеет самопересечений, то каждая прямая пересекает ее дважды; поэтому мы получаем

$$\int n dG = L.$$

Это выражение имеет поразительную общность. Действительно, кривая C может быть овалом, прямоугольником либо «туго закрученной» кривой, которая пересекается прямыми более часто.

Полученное уравнение можно принять за определение длины континуума точек на плоскости⁽⁵⁾. Более того, его можно использовать для построения алгоритма распознавания алфавита образов, имеющих разную длину.

В случае, если кривая C достаточно гладкая, настолько, что можно подсчитать кривизну в точке пересечения случайной прямой с C , тогда случайной переменной будет сумма кривизны во всех таких точках. Нетрудно

¹³Мы пишем $|\sin \theta|$ т. к. все плотности неотрицательны.

¹⁴Заметьте, что n может быть и бесконечностью!

показать, что в этом случае мера всех пересечений будет равна полной кривизне кривой. Наконец, если кривая C не выпуклая и даже незамкнутая, мы получим длину выпуклой оболочки C . Заметим, что полученную меру нетрудно превратить в вероятностную с помощью нормировки, равной длине кривой. Еще больше интересных результатов получится, если ввести меру на парах точек или на отрезках. Все эти меры можно найти в курсах интегральной геометрии [34, 35].

Примечания

1. Группа движений называется транзитивной, если для любой пары точек плоскости можно указать движение, переводящее одну точку в другую. В общем случае, для транзитивной группы должно существовать не больше одной инвариантной меры, с точностью до масштабной константы. Такой меры, однако, может не быть вообще, если группа *сильно транзитивна*, т. е. несколько преобразований группы могут перевести одну точку в другую. Например, добавление к группе \mathfrak{G} растяжений (дилатаций) приводит к сильной транзитивности: на плоскости нет меры, инвариантной относительно движений, поворотов и дилатаций.
2. Дифференциальной формой первого порядка называют любое выражение вида $\omega(u) = \sum a_i(u) dx_i$. Дифференциальной формой второго порядка (или 2-х формой) называют выражение вида $\omega(u) = \sum a_{ij}(u) dx_i \wedge dx_j$, где символ внешнего умножения \wedge определяется в следующем примечании.
3. Формально, пусть R — поле вещественных чисел a, b, c, \dots . L — n -мерное векторное пространство над R с элементами $\alpha, \beta, \gamma, \dots$. Для $p = 0, 1, 2$ построим новые пространства p -векторов $\Lambda^p L$ следующим образом: $\Lambda^0 L = R$, $\Lambda^1 L = L$, $\Lambda^2 L$ — пространство, содержащее суммы $\sum a_i(\alpha_i \wedge \beta_i)$ такие, что

$$\begin{aligned}(a_1\alpha_1 + a_2\alpha_2) \wedge \beta - a_1(\alpha_1 \wedge \beta) - a_2(\alpha_2 \wedge \beta) &= 0, \\ \alpha \wedge (b_1\beta_1 + b_2\beta_2) - b_1(\alpha \wedge \beta_1) - b_2(\alpha \wedge \beta_2) &= 0, \\ \alpha \wedge \alpha &= 0, \\ \alpha \wedge \beta + \beta \wedge \alpha &= 0.\end{aligned}$$

$\alpha \wedge \beta$ называют «внешним произведением» векторов α и β , или бивектором. Пусть e_i — базис в L . Тогда $\alpha = \sum \alpha_i e_i, \beta = \sum \beta_j e_j$ и

$$\alpha \wedge \beta = \sum_{i,j} a_i b_j (e_i \wedge e_j) = \sum_{i < j} (a_i b_j - a_j b_i) e_i \wedge e_j.$$

Величины $e_i \wedge e_j$ образуют базис в $\Lambda^2 L$ с $\dim \Lambda^2 L = \frac{n(n-1)}{2}$. Геометрически, бивектор представляет собой ориентированную площадку, натянутую на пару α, β . Его величина равна площади параллелограмма, то есть $(a_1 b_2 - a_2 b_1)$. В R^3 бивектору $\alpha \wedge \beta$ ставят в соответствие дуальный вектор $\gamma = *(\alpha \wedge \beta)$, который называют векторным произведением.

Внешнее произведение — двухмерный вариант так называемого *альтернированного* или *полярного* произведения трех векторов, которое ввел преподаватель гимназии в Штеттине Г. Грассман (1809–1877), определив его правилами:

$$\begin{aligned} (\vec{v}_1 \vec{v}_2 \vec{v}_3) &= (\vec{v}_2 \vec{v}_3 \vec{v}_1) = (\vec{v}_3 \vec{v}_1 \vec{v}_2) = \\ &= -(\vec{v}_2 \vec{v}_1 \vec{v}_3) = -(\vec{v}_1 \vec{v}_3 \vec{v}_2) = -(\vec{v}_3 \vec{v}_2 \vec{v}_1); \\ (c \vec{v}_1 \vec{v}_2 \vec{v}_3) &= c(\vec{v}_1 \vec{v}_2 \vec{v}_3); \end{aligned}$$

Если v_i — базисные векторы, т. е. любой вектор \vec{w}_j можно записать как

$$\vec{w}_j = \sum a_{ji} \vec{v}_i,$$

то $(\vec{w}_1 \vec{w}_2 \vec{w}_3) = \det \|a_{ij}\| (\vec{v}_1 \vec{v}_2 \vec{v}_3)$, где детерминант равен объему параллелепипеда, натянутого на три вектора.

4. Ее называют длиной Фавара и измеряют экспериментально следующим образом. Возьмем решетку, образованную семейством равноудаленных параллельных линий G_i ($i = 1, 2, \dots$). Допустим, что подлежащая измерению дуга C пересекает G_i в n_i точках так, что $s_0 = \sum n_i$ есть общее число точек пересечения. Вращая решетку на угол $(k/m)\pi$ ($k = 0, 1, \dots, m-1$), получим s_k пересечений и общая сумма будет равна $N = \sum_{k=0}^{m-1} s_k$. Обозначая расстояние между G_i и G_{i+1} через a , можно рассматривать величину $(N\pi a)/(2m)$ как приближенную оценку длины C .

Путеводитель по литературе. Отличным введением в интегральную геометрию являются две книги Л. Сантало на русском языке [34, 35]. Некоторые результаты из этой области, так же как и краткое историческое введение в дифференциальные формы, можно найти в учебнике [36]. Великолепная статья А. Новикова [31] дает неформальное введение в предмет. Наконец, хорошим учебником по теории дифференциальных форм является [37].

Математическая морфология: Функционалы Минковского

Округлость, постигаемая взором,
допускает увеличение и
уменьшение: одно кругло, другое
круглее . . . Все округлое округло в
силу причастности к ней.

Н. Кузанский
«Игра в шар»

Конечное множество точек $\{x_i | i = 1, 2, \dots, N\} \in R^d$ имеет тривиальную дискретную топологию. Если число точек невелико, статистические критерии не применимы и, следовательно, мы не можем сравнить две точечные выборки. Прием, который позволяет это сделать, аналогичен алгоритму «степного пожара». «Декорируем» каждую точку x_i шаром $B_r(x_i) = \{x \in R^d | \|x - x_i\| \leq r\}$ радиуса r . Набор шаровых окрестностей образует покрытие Минковского⁽¹⁾: $B(r) = \bigcup_{i=1}^N B_r(x_i)$. При синхронном увеличении радиусов шаров часть окрестностей начнет объединяться. Следовательно, будет меняться число связанных компонент, сумма периметров окрестностей и их суммарная площадь. Все эти величины, параметризованные радиусом шара, можно использовать для описания формы и содержания дискретного множества. Необходимый формализм основан на так называемых *функционалах Минковского*.

Рассмотрим компактное выпуклое тело $K \in R^3$ с регулярной¹⁵ границей $\partial K \in C^2$. Выпуклое тело K_ε называют телом *параллельным* K на расстояние ε . Оно образовано объединением всех ε -шаров, содержащих

¹⁵Т. е. граница имеет касательную и нормаль.

точки из K и всех точек, удаленных от него на расстояние не превосходящее ε :

$$K_\varepsilon = \bigcup_{x \in K} B_\varepsilon(x).$$

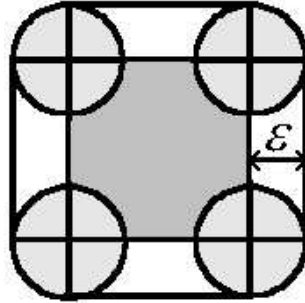


Рис. 13. Построение параллельного тела для квадрата

Обычные евклидовы меры (площадь F_ε и объем V_ε) параллельного тела можно выразить полиномами по степеням ε . Например, для квадрата A со стороной a (рис. 13), «параллельная» площадь выражается полиномом:

$$F_\varepsilon(A) = a^2 + 4a\varepsilon + 4(\pi/4)\varepsilon^2.$$

Аналогично, для куба Q , с тем же ребром, легко получить:

$$V_\varepsilon(Q) = a^3 + 6a^2\varepsilon + 12a(\pi/4)\varepsilon^2 + 8(4\pi/24)\varepsilon^3.$$

Наиболее общее выражение для V_ε в d -мерном пространстве дает формула Штейнера⁽²⁾:

$$V_\varepsilon(K) = \sum_{i=0}^d \binom{d}{i} W_i(K) \varepsilon^i,$$

где $W_i(K)$ называют функционалами Минковского.

Для $d = 3$ функционалы Минковского W_i имеют вид [39]:

$W_0(K) = V(K)$ — евклидов объем K ,

$3W_1(K) = F(K)$ — евклидова площадь K ,

$3W_2(K) = H(K)$ — интегральная кривизна границы ∂K ,

$3W_3(K) = G(K) = 4\pi\chi(K)$ — гауссова кривизна границы ∂K , где χ — эйлерова характеристика тела K , которая будет рассмотрена ниже.

Для $d = 2$ существует всего три функционала Минковского, которые полностью описывают дискретное множество точек на плоскости для любого значения радиуса шаров покрытия:

$W_0 = F$ — суммарная площадь области, занятой кружками,

$W_1 = U/2$ — периметр всех окрестностей,

$W_2 = \chi/\pi$ — число односвязных областей минус число дыр в них.

Функционалы W_i обладают следующими важными свойствами:

- **Аддитивность:** $W_i(A \cup B) = W_i(A) + W_i(B) - W_i(A \cap B)$.
- **Инвариантность.** Пусть \mathfrak{G} — группа движений на плоскости и gA ($g \in \mathfrak{G}$) — образ множества A после преобразования. Тогда: $W_i(gA) = W_i(A)$.
- **Непрерывность.** Для сходящейся последовательности⁽³⁾ выпуклых множеств $A_n \rightarrow A$, $n \rightarrow \infty$: $W_i(A_n) \rightarrow W_i(A)$.

Характеристика Эйлера

Характеристика Эйлера играет особую роль. Поэтому рассмотрим ее подробнее. Прежде всего заметим, что пересечения шаров образуют не только выпуклые тела. Рассмотрим класс \mathfrak{B} всех подмножеств $B_i \in R^2$, которые можно представить как конечное объединение выпуклых множеств A_i , т. е.

$$B_i \in \mathfrak{B}, \text{ если } B_i = \bigcup_{i=1}^N A_i; \emptyset \in \mathfrak{B}.$$

Тогда можно ввести χ как аддитивный функционал¹⁶ на \mathfrak{B} , так что для $A, B \in \mathfrak{B}$:

$$\chi(A \cup B) = \chi(A) + \chi(B) - \chi(A \cap B)$$

и для любого выпуклого множества A

$$\chi(A) = \begin{cases} 1, & A \neq \emptyset, \\ 0, & A = \emptyset. \end{cases}$$

¹⁶Его часто называют функционалом Хадвигера.

Рассмотрим, например, кольцо, которое не является выпуклым множеством, но его можно представить объединением трех дисков уложенных на плоскости. Тогда из условия аддитивности сумма положительных слагаемых (три диска) равна трем. Три точки взаимных пересечений дисков дадут значение (-3) . В итоге, характеристика Эйлера для кольца равна нулю.

Приведенное определение χ согласуется с классической формулой Эйлера: $(\text{число вершин}) - (\text{число ребер}) + (\text{число граней}) = 1$, для любого замкнутого полигона на плоскости. Для ее применения к диску достаточно деформировать последний в квадрат. Формула Эйлера лежит в основе компьютерного алгоритма, принадлежащего Серра [38], для вычисления χ на бинарных изображениях (рис. 14).

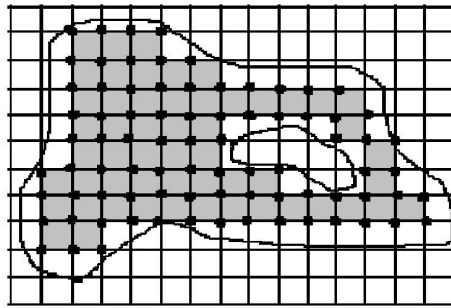


Рис. 14. Алгоритм Серра для вычисления характеристики Эйлера

Для получения χ достаточно подсчитать число вершин, ребер и граней клеток, которые целиком содержатся в области, ограниченной контурами на рисунке, и применить затем формулу Эйлера.

Опишем еще один алгоритм, который позволяет вычислить χ в пространстве любой размерности. В нем характеристика Эйлера определяется по индукции. Начнем со случая, когда $X \subset R^1$ и состоит из конечного объединения n несвязных сегментов. Поскольку каждый сегмент является элементом \mathfrak{B} , очевидно, что $\chi = n$. Рассмотрим теперь случай R^2 (рис. 15).

Обозначим через $\Delta(x)$ подвижную прямую линию перпендикулярную оси x . При своем движении слева направо она будет «сканировать» множество X и поможет нам вычислить некоторую вспомогательную величину $h(x)$. Последняя является разностью функционалов Хадвигера при двух

положениях $\Delta(x)$, отличающихся на величину ε :

$$h(x) = \lim_{\varepsilon \rightarrow 0} \left\{ \varphi[X \cap \Delta(x)] - \varphi[X \cap \Delta(x + \varepsilon)] \right\}.$$

Очевидно, что $h(x) \neq 0$ только в конечном числе точек x_i , в которых $\Delta(x)$ касается границы ∂X множества X . Определим теперь эйлерову характеристику как

$$\chi(X) = \sum_i h(x_i).$$

В качестве примера вычислим $h(x)$ для трех множеств X_1, X_2, X_3 показанных на рис. 15. Очевидно, что в точке 1 для всех множеств мы получим $h(1) = 0$. Для множества X_1 имеем: $h(2) = 1$ и, следовательно, $\chi(X_1) = 1$. Множество X_2 имеет четыре точки касания, для которых $h(2) = -1, h(3) = +1, h(4) = +1$; поэтому $\chi(X_2) = 1$. Наконец, для кольца X_3 мы имеем значения $h(2) = -1, h(3) = 0, h(4) = +1$, которые в сумме дают $\chi(X_3) = 0$. Интересно отметить, что $h(x) = +1$, если ∂X в точке касания с $\Delta(x)$ локально выпуклая, т. е. находится слева от прямой, и $h(x) = -1$, если граница локально вогнута в x . Приведенный алгоритм допускает обобщение на произвольную размерность; например, в R^3 «сканирующим» элементом будет уже не прямая, а плоскость. Пример с множеством X_3 приводит к «наивному» алгоритму для вычисления χ на бинарных изображениях: подсчитаем общее число кластеров (островов), образованных окрашенными пикселями и вычтем из полученной суммы число «дыр» в них.

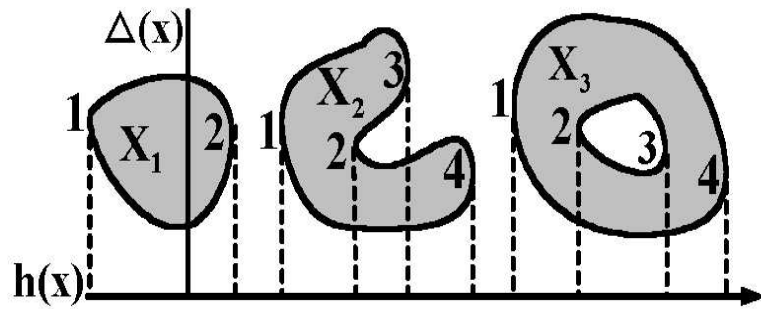


Рис. 15. Определение χ в R^2

Характеристику Эйлера можно ввести в контексте дифференциальной геометрии. Рассмотрим замкнутую гладкую кривую C , параметризованную длиной дуги s . В каждой точке такой кривой существует касательная и нормаль. При переходе к соседней точке касательная поворачивается на некоторый угол θ , зависящий от кривизны ⁽⁴⁾. *Геодезическая кривизна* кривой измеряется скоростью изменения этого угла: $k_g = d\theta/ds$. Интеграл полной кривизны определяет эйлерову характеристику кривой согласно формуле

$$2\pi\chi = \oint_C k_g ds.$$

Рассмотрим теперь двумерное многообразие M с (одномерной) кусочно-гладкой границей ∂M . Знаменитая теорема Гаусса-Бонне [42, 60] говорит, что

$$\sum_{i=1}^n (\pi - \alpha_i) + \int_{\partial M} k_g ds + \int_M K dA = 2\pi\chi(M)$$

где $\alpha_1, \alpha_2, \dots, \alpha_n$ — внутренние углы при вершинах на границе ∂M ; k_g — геодезическая кривизна кривой ∂M ; K — гауссова кривизна M ; ds и dA — элементы длины и площади соответственно. Таким образом, эта теорема связывает локальные (кривизна) и глобальные (χ) свойства многообразий. Заметим, наконец, что все функционалы Минковского могут быть выражены через χ [38, 41].

Характеристика Эйлера и персептроны

М. Минский и С. Пейперт [43] исследовали характеристику Эйлера в связи с геометрическими множествами, которые может распознать персептрон. Уместно привести здесь несколько интересных результатов. Напомним, прежде всего, что «предикатом» называется переменное высказывание, т. е. функция $\psi(X)$ от множества X , истинность (1) или ложность (0) которого зависит от выбора X . Например, предикат «связность» определяется как $\psi_{\text{связность}}(X) = 1$, если X — связная фигура и $\psi_{\text{связность}}(X) = 0$ в противном случае. Предикат ψ называется *конъюнктивно локальным k -го порядка*, если его значение зависит от k тривиальных предикатов, например, предикатов принадлежности. Известно, например, что выпуклость фигуры X можно проверить, используя тройки точки p, q, r . Три тривиальных предиката $\varphi(p) = \varphi(q) = \varphi(r) = 1$, если эта тройка принадлежит X . В случае, если какой-либо из этих предикатов равен нулю, X не является выпуклым.

Таким образом, конъюнктивно локальный предикат $\psi_{\text{выпуклость}}(X)$ будет истинным, если все три предиката принадлежности $\varphi = 1$. Очевидно, он имеет третий порядок, т. е. решение о том, является ли X выпуклым принимается при «параллельном» вычислении предикатов φ тремя нейронами, каждый из которых считывает отдельную точку. Такой предикат может быть реализован с помощью персептрона ограниченного диаметра. Однако, предикат $\psi_{\text{связность}}(X)$, как доказано в [43], не является конъюнктивно локальным какого-либо порядка. Иными словами, его порядок с ростом X неограниченно возрастает и он не может быть реализован персептроном ограниченного диаметра. Этот результат согласуется с интуицией, потому что связность — это глобальное понятие, и невозможно выяснить, является ли фигура связной, располагая только локальной информацией. Замечательно, что предикат, вычисляющий характеристику Эйлера, т. е. предикат $\psi_{\text{Эйлер}}(X) = (\text{число компонент}(X)) - (\text{число дыр}(X))$ является конъюнктивно локальным конечного порядка.

Примечания

1. *Герман Минковский* (1864–1909) — выдающийся математик, родился в России. Был профессором в Бонне, Кенигсберге, Цюрихе и Геттингене. Создал особое учение о «геометрии чисел» («*Geometrie der Zahlen*»; «*Diophantische Approximationen*» и др.). Его лекции по математике слушал *А. Эйнштейн*. В своей последней работе: «*Raum und Zeit*» (1909) Минковский дал математическую формулировку так называемого «принципа относительности».
2. *Штейнер Якоб* (1796–1863) — немецкий математик. Родился в Утценсторфе (Швейцария). Член Берлинской Академии наук. Один из творцов проективной геометрии. В основной своей работе «Систематическое развитие зависимости геометрических образов одного от другого» (1834) построил геометрию, не используя аналитические методы. Автор книг «Геометрические построения, осуществляемые с помощью прямой и постоянного круга» (1833), «О наибольших и наименьших значениях плоских фигур и о сфере» (1842).
3. Определение сходимости последовательности геометрических множеств требует специальной метрики, которая позволяет вычислить расстояние между двумя объектами. В качестве такой метрики ис-

пользуется метрика Хаусдорфа [40]:

$$h(A, B) = \inf \{ \varepsilon : A \subset B_\varepsilon \text{ and } B \subset A_\varepsilon \}.$$

Иными словами, строим параллельное тело B_ε , «раздувая» B посредством ε -шаров так, чтобы полученный образ «поглотил» множество A . Такая ситуация будет соответствовать некоторому значению ε_1 . Прделаем теперь то же самое с множеством A и получим значение ε_2 . Минимальное из них и будет расстоянием Хаусдорфа.

4. Кривизной кривой в точке M_0 называется предел отношения угла $\Delta\varphi$ поворота касательной на дуге $\Delta s = M_0M$ к длине этой дуги, при $M \rightarrow M_0$: $k = \lim_{M \rightarrow M_0} \Delta\varphi / \Delta s$. В общем случае поверхность имеет в каждой точке две кривизны k_1 и k_2 . Полная (гауссова) кривизна определяется выражением $K = k_1 k_2$. Для сферы, например, $k_1 = k_2 = 1/R$, поэтому $K = 1/R^2$ и интеграл полной кривизны $\oint K dA = 1/R^2 4\pi R^2 = 4\pi$.

Путеводитель по литературе. Лучшим пособием по морфологии является монография Серра [38], но ее трудно найти. Хорошее введение в предмет содержат обзорные статьи [39, 40], доступные на веб-страницах. Полезной является статья [41]. Топологическое определение эйлеровой характеристики приведено в учебнике [10]; связь с топологией и дифференциальной геометрией лучше всего описана в книге [42]. В монографии [43] приведены предикаты связности и их связь с персептронами.

Стохастическая геометрия: Контурная статистика

Культурный человек только слегка обгрызает кости, а потом бросает их под стол.

*«Мысли людей великих, средних и
песика Фафика»*

Математическая морфология позволяет получить геометрические характеристики паттернов, такие как площадь, периметр, число связных компонент и др. Когда они не доступны прямым измерениям, оценки этих

величин можно получить косвенно, используя методы интегральной геометрии. Множества, которыми она занимается, полагают обычно детерминированными — случайным является только их положение относительно наблюдателя. Однако возможна и другая ситуация, когда случайными являются сами множества. На рис. 16 приведена томограмма алюминиевой пены. Геометрическая сложность такой пористой структуры настолько ве-

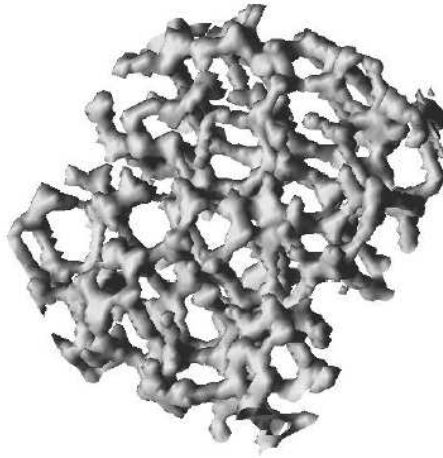


Рис. 16. Алюминиевая пена

лика, что вынуждает нас перейти к статистическому описанию. Иными словами, мы предполагаем, что элементы этой структуры, заполненные веществом¹⁷, так же как и поры между ними имеют случайную форму. Более того, мы будем считать случайной и структуру, полученную сечением нашего объекта произвольной плоскостью. С такими паттернами постоянно приходится иметь дело в металлографии, петрографии, пористых средах, ветровых волнах и биологии. Оказывается, что распределение вещества во Вселенной также имеет аналогичную «топологию губки»⁽¹⁾.

Математическое определение случайного множества слишком сложное, и я не буду его приводить¹⁸. Случайные формы и случайные множества придают вероятностную «окраску» функционалам Минковского, которые

¹⁷ Их называют «гранулы».

¹⁸ Его впервые дал французский математик Г. Шоке в 1953 году.

становятся теперь «статистиками». Я покажу, как получаются такие статистики на примере игрушечной модели. Представим себе, что в кубе случайным образом распределено конечное множество сфер разных размеров. Оценим суммарную площадь поверхности всех сфер и поделим эту величину на объем куба. То, что получится, называют статистикой N_3 . Выберем случайную плоскость, пересекающую куб. Мы увидим на ней множество окружностей разных размеров — следы разрезанных сфер. Оценим суммарный периметр всех кругов, приходящийся на единицу площади сечения. Следуя канонам интегральной геометрии эту величину, разумеется, надо усреднить по множеству всех сечений. Полученное значение называют статистикой N_2 . Выберем наконец случайную прямую на секущей плоскости. Она пересечет некоторые из окружностей, и результатом будет конечное множество точек на прямой. Их плотность, т. е. число точек на единицу длины прямой, даст нам последнюю статистику — N_1 . Конечно, как и в предыдущем случае, последнюю оценку надо усреднить по множеству всех случайных прямых. В нашем примере случайные множества сферически симметричны. Поэтому можно связать все три N_i , $i = 1, 2, 3$ простыми выражениями, которые являются функционалами Минковского в «статистическом облике». Наша модель иллюстрирует основной прием *стохастической геометрии* — использование геометрических характеристик для описания статистических свойств предельно сложных паттернов. Я приведу один замечательный результат из этой области [47, 49].

В физике для обозначения пространственных паттернов чаще используют термин «случайное поле», имея ввиду стохастический векторный процесс $X(t) \in R^n$, где $t \in R^k$. Реализацией $X(t)$ может быть, например, высота морской волны ($n = 1$) или напряженность магнитного поля на поверхности Солнца ($n = 3$), которые измеряются в точке с прямоугольными или сферическими координатами ($k = 2$). В большинстве приложений интерес представляют оценки вероятности того, что максимумы случайного поля $\sup X(t)$, измеренного с пространственным разрешением λ , превышают некоторый порог u : $Prob(\sup X(t) \geq u)$. Иными словами, нас интересует статистика максимальных выбросов¹⁹ поля выше некоторого заданного сечения, подобного тому, что мы использовали в игрушечной модели. К сожалению, не существует аналитических выражений для такой вероятности в случае произвольного поля. Но, оказывается, что по меньшей мере для гауссовских полей искомая вероятность выражается через

¹⁹Иногда говорят «экскурсий».

математическое ожидание $E(\chi)$ эйлеровой характеристики экскурсий поля [47]:

$$\begin{aligned} Prob(\sup X(t) \geq u) \propto E(\chi) = & \frac{\lambda^3}{(2\pi)^2} (u^2 - 1) \exp(-u^2/2) + \\ & + \frac{(1/2)\lambda^2}{(2\pi)^{3/2}} u \exp(-u^2/2) + \frac{2\lambda}{2\pi} \exp(-u^2/2) + \\ & + \frac{\chi}{(2\pi)^{1/2}} \int_0^\infty \exp(-z^2/2) dz. \end{aligned}$$

В случае R^3 в сечении рельефа поля мы будем получать контуры. Поэтому статистику N_2 часто называют «контурной». Ее истоками является задача о выбросах случайной функции в R^1 , которую исследовал С. Райс [45]. Рассмотрим одномерный случайный процесс $\xi(t)$, $t \in [a, b]$, и его график вместе с горизонтальной прямой $\xi = h$ (рис. 17). Множество выбросов («экскурсий») графика за уровень h называется множество аргументов, для которых $\xi(t)$ равна или превышает h , т.е. $A_\xi(h) = \{t | \xi(t) \geq h\}$. В общем случае множество $A_\xi(h)$ состоит из отдельных точек и интервалов.

Для реализации выброса в точке t необходимо одновременное выполнение двух условий: $\xi(t) < h$; $\xi(t + dt) > h$, где dt — интервал. Реализация выброса в интервале dt эквивалентна выполнению двойного неравенства:

$$h - \xi'(t)dt < \xi(t) < h; \quad \xi'(t) > 0,$$

где производная $\xi'(t)$ случайного процесса понимается в стохастическом смысле.

Вероятность выброса $Prob[h - \xi'(t)dt < \xi(t) < h]$ можно выразить через совместную плотность распределения $\xi(t)$ и ее производной в одной и той же точке. Если скорость изменения ординаты и сама ордината являются независимыми переменными²⁰, можно «расцепить» совместную плотность на два сомножителя и получить простые выражения для среднего числа выбросов и протяженности одного выброса, т.е. средней величины интервала Δt , в котором функция $\xi(t)$ пребывает выше уровня h . Эти выражения и были получены С. Райсом в 1944 году для стационарного гауссовского процесса.

Пусть теперь $X(t)$ — непрерывная функция на $[0, T]$, не равная тождественно h в данном интервале, и $X(0) \neq X(T) \neq h$. Будем говорить,

²⁰Это будет, например, если процесс гауссовский.

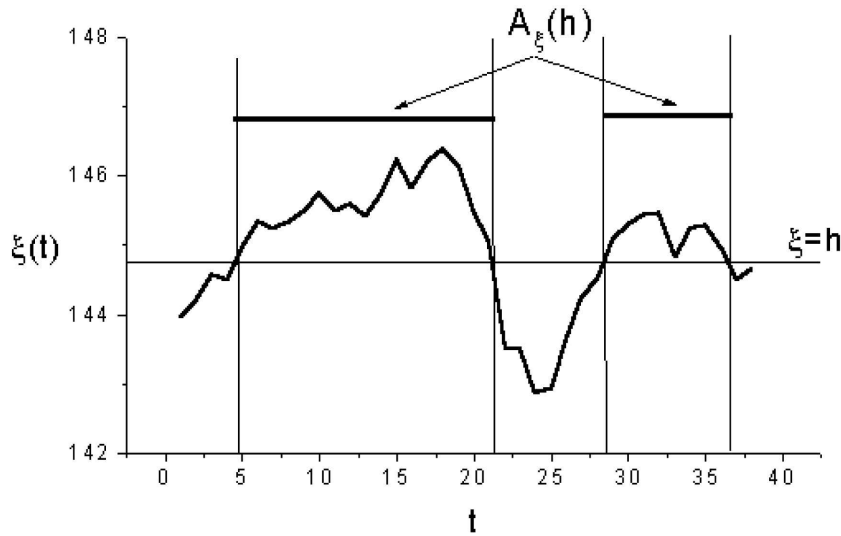


Рис. 17. Задача о выбросах случайной функции

что $X(t)$ пересекает уровень h снизу вверх в точке t_0 , если существует такое $\varepsilon > 0$, что $X(t) \leq 0$ в $(t_0 - \varepsilon, t_0)$ и $X(t) \geq 0$ в $t_0, (t_0 + \varepsilon)$. Число этих точек t_0 в $[0, T]$ назовем числом пересечений снизу вверх уровня h и обозначим $N_h(X, T)$. Связь между $N_h(X, T)$ и множеством выбросов $A_h = A_h(X(t)) = \{t \in [0, T] : X(t) \geq h\}$ очень простая. Если $X(0) < h$, то число пересечений и число компонент A_h равны; в случае $X(0) \geq h$, $A_h = N_h + 1$.

Обобщение множества экскурсий на случай полей должно включать случайные точки вида:

$$\{t \in S : X(t) = h\} = \partial A_h(X, S), \quad t \in R^N, \quad N > 1,$$

где ∂A_h означает границу множества A_h . Для R^2 , например, ∂A_h — это границы контуров на плоскости. Понятно, что прямое обобщение для числа пересечений в этом случае невозможно. Непонятно, например, как различать в этом случае пересечения снизу вверх и наоборот. Соответствующий формализм для этих ситуаций был создан Р. Адлером [47, 48]. Его основная идея заключается в том, чтобы статистику выбросов свести к статистике

функционалов Минковского. Мы рассмотрим ниже два примера нашего опыта работы с контурной статистикой. Первый из них относится к радиоэкологии бывшего Семипалатинского ядерного полигона (СИЯП), второй взят из области физики глобального магнитного поля Солнца.

Приложение методов контурной статистики

Я начну с первого примера. На территории СИЯП за период 1949–1989 годов было произведено 470 ядерных взрывов разной мощности (90 воздушных, 25 наземных и 354 подземных). Их суммарное воздействие и влияние сопутствующих климатических, гидрологических и техногенных факторов сформировали структурированное стохастическое поле загрязнений, содержащее широкий диапазон долгоживущих радионуклидов. Наиболее опасными из них являются: ^{14}C (период полураспада 5730 лет), ^{60}Co (5.26 лет), ^{90}Sr (28.5 лет), ^{137}Cs (30 лет), ^{239}Pu (24113 лет), ^{241}Am (14.5 лет). Эти техногенные изотопы наложены на естественный фон, вызванный природной радиоактивностью горных пород, почвы, воды и воздуха. Современная радиационная обстановка на СИЯП до сих пор изучена недостаточно²¹. Чтобы понять, в чем тут дело, необходимо сказать кое-что о методах измерений. Поверхностное загрязнение радионуклидами измеряется двумя способами. Первый из них — это отбор проб. На стандартной квадратной площадке, скажем размером один 1 м^2 , из вершин квадрата и его центра берутся образцы почвы. Они перемешиваются в лаборатории и делается радиоизотопный анализ смеси. Полученное содержание изотопов приписывается этой площадке. Понятно, что этот метод трудно использовать для диагностики больших территорий: площадь только самого полигона превышает 16000 кв. км.

Второй метод использует авто- или аэрогамма-спектрометрическую съемку территории по сетке параллельных маршрутов. Для типичной съемки в масштабе 1:1000000 расстояния между соседними маршрутами составляли 10 км. К самолету или вертолету подвешивается гондола с многоканальным гамма-спектрометром. Его расстояние от поверхности земли не должно быть значительным, что само по себе уже проблема для полетов по пересеченной местности. Интерполяция измерений полученных такой редкой сетью вызывает большие трудности. Дело в том, что радионуклид-

²¹Исследовано менее 1% загрязненной площади.

ные загрязнения имеют мультифрактальный скейлинг²² и поэтому, правомерность применения популярных аппроксимационных алгоритмов весьма проблематична. Используя специальные методы, основанные на фрактальной интерполяции, можно получить более или менее приемлемую карту поверхностных загрязнений. Она не слишком хорошо отображает метрические характеристики поля, но сохраняет топологическую информацию о статистике выбросов. Это очень важно для практических задач. Смежные с полигоном территории хотелось бы использовать в сельском хозяйстве. Для этого необходимо иметь представление не только о плотности распределение пиков загрязнения, но и уметь ответить на вопрос: какова вероятность встретить радионуклидное пятно цезия с интенсивностью в k раз превышающей предельно допустимую норму? Вот для таких задач контурная статистика оказалась идеальным инструментом. На рис. 18 в качестве примера приведена пиксельная карта канала Cs для одного из фрагментов СИЯП. Каждый пиксел имеет размер 20×50 м. Сам участок называют полигоном «Иртыш»; он был создан для калибровки приборов и поэтому измерялся наиболее тщательно⁽²⁾. Аналогичные карты имеются для каналов природных изотопов K , Th , U .

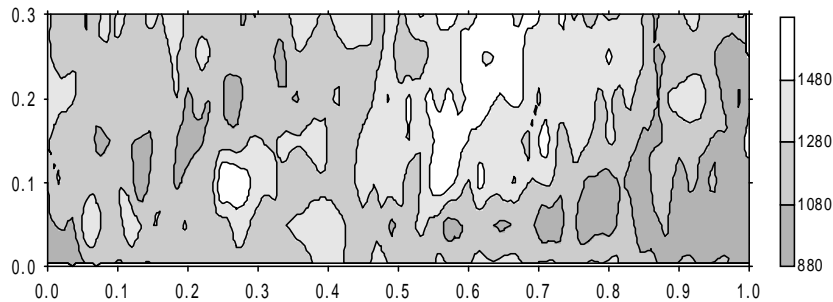


Рис. 18. Пиксельная карта канала Cs для площадки «Иртыш»

Для каждой карты были получены сечения уровня поля в единицах дисперсии. На рис. 19 приведена интегрально-геометрическая характеристика Эйлера⁽³⁾ $\langle \chi_{IG} \rangle$ для всех каналов вместе с теоретической кривой для модели гауссовского поля. Видно, что все каналы разделяются по поведению функционала и $\langle \chi_{IG} \rangle$ для цезия Cs больше всего отличается от модели

²²Иерархическую структуру пятен.

случайного поля. Эти результаты использовались для получения экологических оценок. Подробности можно найти в наших работах [52, 53].

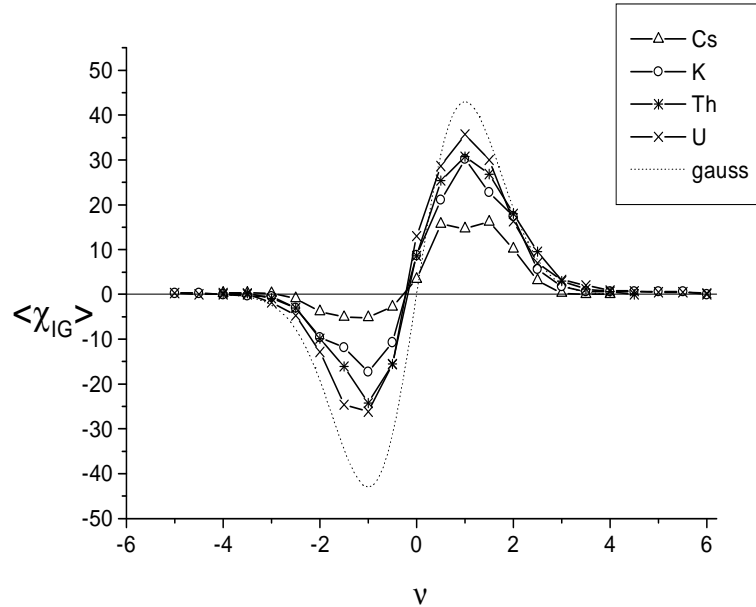


Рис. 19. Графики характеристики Эйлера для различных изотопов

Рассмотрим пример из другой области. Известно, что солнечное магнитное поле имеет сложную пространственную структуру и квазициклическое поведение во времени. Это поле продуцируется диффузными компонентами и компактными магнитными структурами (пятнами) разных размеров и с различным временем жизни. В первом приближении его можно рассматривать как структурированное случайное поле⁽⁴⁾. На больших масштабах магнитные структуры можно рассматривать как фоновое магнитное поле. Это крупнозернистое поле проявляется в распределении униполярных областей, которые эволюционируют с характерным временем около 11 лет. Линии раздела полярностей изображаются на цилиндрической развертке солнечной поверхности в форме так называемых $H - \alpha$ синоптических магнитных карт (рис. 20). Такие карты дают усредненную динамику

поля Солнца во временном окне около 27 дней²³. Доступная последовательность синоптических карт начинается с 1915 года. Более детальная информация о магнитном поле содержится в синоптических картах Wilcox Solar Observatory, которые существуют с 1976 года и содержат значения самого поля, а не только его знак.

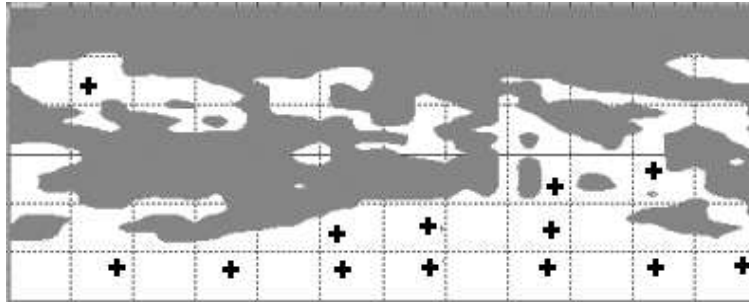


Рис. 20. Пример $H - \alpha$ синоптической карты. Знак + соответствует северной полярности.

До недавнего времени синоптические карты использовались только для феноменологического анализа.

Заметим, что такие карты представляют реальные магнитные структуры по меньшей мере гомеоморфно так, что топологические свойства обычно сохраняются. Следовательно, можно извлечь физически интересную информацию, используя методы контурной статистики. В качестве примера на рис. 21 приведен график сглаженной эйлеровой характеристики, полученной по $H - \alpha$ картам для восьми Солнечных циклов. Для сравнения на том же графике изображены сопутствующие числа Вольфа²⁴. Полученный временной ряд χ использовался для независимой оценки размерности «солнечного аттрактора». Оказалось, что другой функционал Минковского — периметр линии раздела полярностей полезен для прогноза солнечных вспышек. Его изменения предшествуют увеличению так называемого вспышечного индекса. Подробности можно найти в нашей работе [55].

Примечания

²³Один оборот Солнца.

²⁴Они пропорциональны числу солнечных пятен.

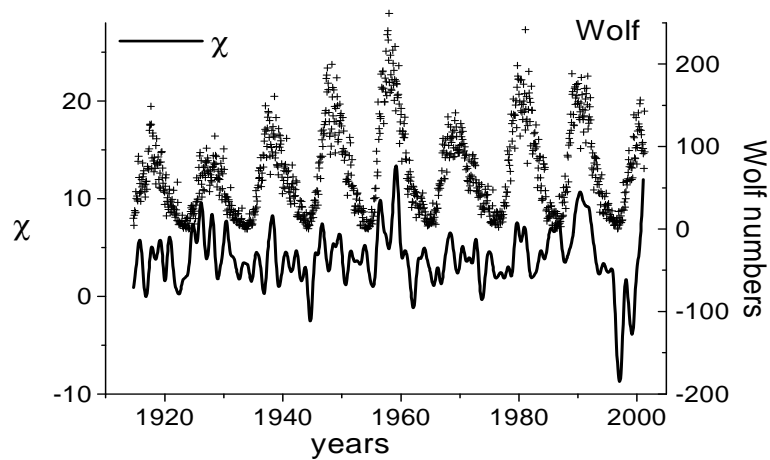


Рис. 21. Эйлерова характеристика последовательности синоптических карт для восьми Солнечных циклов

1. Такое распределение получится, если мы перейдем от дискретного наблюдаемого распределения вещества в галактиках и их скоплениях к непрерывному распределению. Его можно получить если «размазать» наблюдаемое вещество по ячейкам взвешенной диаграммы Вороного, а затем сгладить плотность на ребрах, используя, например, гауссовский фильтр.
2. Полигон «Иртыш» расположен в зоне «Опытного поля» СИЯП, вблизи эпицентра первого атомного взрыва, и состоит из трех площадок размером 10×0.4 км. Аэро-гаммаспектрометрическая съемка масштаба 1 : 5000 регистрировала полный спектр жесткого гамма-излучения в диапазоне 0.2 – 3.0 МэВ с разбивкой диапазона на 256 уровней. Высота залетов 20 м; средняя скорость вертолета — около 80 км/ч.
3. Интегрально-геометрическая характеристика Эйлера $\langle \chi_{IG} \rangle$ отличается от стандартной учетом влияния границ карты. Выберем начало координат в левом нижнем углу карты. При подсчете χ не будем учитывать контуры, имеющие общие точки с координатными осями; это будет величина χ_{IG} . После переноса системы координат в следующий угол карты получим новую оценку χ_{IG} . Усреднение этих оценок

по всем углам карты дает $\langle \chi_{IG} \rangle$, которая может принимать дробное значение.

4. Магнитогаммы, полученные с высоким разрешением, показали, что магнитный поток имеет перемежаемую мультифрактальную структуру в широком диапазоне масштабов.

Путеводитель по литературе. Монография Ж. Матерона [44], посвященная стохастической геометрии, рассчитана на математиков. Задача о выбросах описана в учебнике [45]; много полезного можно найти в обзоре [46], посвященном ветровым волнам. Классическим пособием по теории случайных полей является монография Р. Адлера [47]. Более доступны главы неопубликованной пока книги [48]. Для физиков более интересны статьи К. Ворслея [49–51]. Лучше начать с его популярной статьи [49]. В наших работах описаны некоторые приложения функционалов Минковского к радиоэкологии [52, 53], к сейсмологии [54] и физике Солнца [55]. Наиболее популярны приложения морфологических функционалов в космологии. Интересные результаты содержатся в работах [56–58]; в [59] можно найти интересную попытку введения метрики на множестве астрофизических карт, основанную на морфологии.

Вычислительная топология: Гомологии и числа Бетти

О! О! О! О!

В. Шекспир
«Отелло», акт V, сцена 2

Функционалы Минковского описывают главным образом геометрию паттерна. Однако два объекта, имеющие идентичную геометрию, могут иметь совершенно различную топологию. Простой пример такой ситуации — два фрактала, отличающиеся числом «дыр». Площади фрагментов, наделенных мерой, и их периметры совпадают. Более того, совпадают и фрактальные размерности, потому что соответствующие алгоритмы учитывают только число непустых боксов. Различить два пористых паттерна можно только в рамках *вычислительной топологии*. Эта совсем молодая наука изучает алгебраические инварианты элементарных множеств (топологических клеток), составляющих паттерн. Я ограничусь здесь только некоторыми

исходными понятиями *алгебраической топологии* и покажу их связь с дифференциальной топологией многообразий.

Отрезок соединяет две точки, на три неколлинеарные точки можно «натянуть» треугольник; четыре некомпланарные точки служат вершинами тетраэдра (см. рис. 22). Эти выпуклые множества называют симплексами.

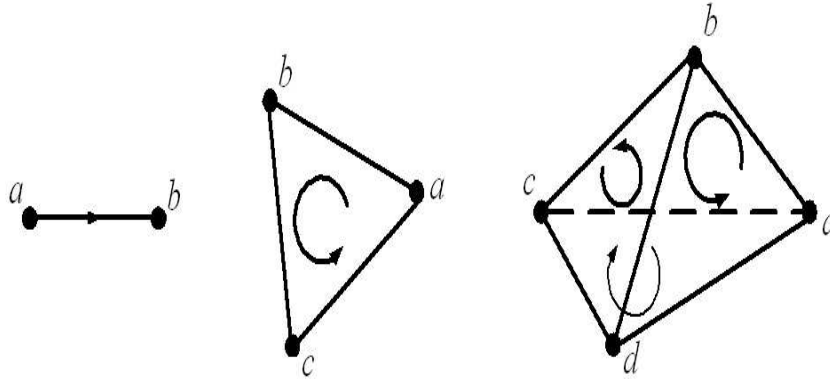


Рис. 22. Примеры симплексов

Они различаются размерностями и обозначаются перечислением вершин, задающих ориентацию: 0-симплекс — точка, 1-симплекс — отрезок $[a, b]$, 2-симплекс — треугольник $[a, b, c]$, 3-симплекс — тетраэдр $[a, b, c, d]$. Алгебраически ориентация задается знаками при вершинах (границах) симплексов, например, $[a, b] = b - a$. Линейная комбинация

$$c^k = \sum a_i \Delta_i^k$$

симплексов Δ_i^k называется *цепью* или *k-цепью*. Здесь $a_i \in Z$ — целые числа и все симплексы в цепи имеют размерность k . Как известно, множество целых чисел Z образует абелеву (коммутативную) группу относительно сложения. Цепи также образуют абелеву группу; обозначим ее как C_k . Границей $\partial \Delta^k$ симплекса $\Delta^k = [a_0, a_1, \dots, a_k]$ являются, очевидно, Δ^{k-1} -симплексы. Они представляют собой сумму всех его $(k-1)$ -мерных граней с индуцированной ориентацией:

$$\partial_k [a_0, a_1, \dots, a_k] = \sum_{i=0}^k (-1)^i [a_0, \dots, \widehat{a_i}, \dots, a_k],$$

где \widehat{a}_i — означает отсутствие i -ой вершины в сумме. Вычислим, например, границы для первых двух симплексов на рис. 22:

$$\begin{aligned}\partial_1[a, b] &= b + (-1)^1 a, \\ \partial_2[a, b, c] &= [b, c] - [a, c] + [a, b] = [b, c] + [c, a] + [a, b].\end{aligned}$$

Оператор ∂ является линейным. Его применение к цепям определяет *гомоморфизм*²⁵: $\partial_k : C_k \rightarrow C_{k-1}$. Последовательное применение ∂ понижает размерность цепи и приводит к *цепному комплексу* на группах цепей:

$$\dots \rightarrow C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \rightarrow \dots \rightarrow C_0 \xrightarrow{\partial_0} 0.$$

Важным является свойство: $\partial^2 c = \partial \partial c = 0$, т. е. граница границы есть нуль. Проверим его для 2-симплекса:

$$\partial_1 \partial_2[a, b, c] = [c] - [b] - [c] + [a] + [b] - [a] = 0.$$

Цепь z называют *циклом*, если ее граница равна нулю: $\partial z = 0$. Множество k -мерных циклов $Z_k \subset C_k$ образует подгруппу в группе цепей. Элементы группы C_k , которые при гомоморфизме ∂ отображаются в нуль, называют *ядром* ($\ker \partial$) гомоморфизма, так что $Z_k = \ker \partial$. Цепь b называют *границей*, если ее можно представить в виде границы некоторой $(k+1)$ -мерной цепи h , т. е. $b = \partial_{k+1} h$. Например, окружность (1-цепь) является границей площадки внутри нее (2-цепи). Поскольку $\partial_k b = \partial \partial_{k+1} h = 0$, множество границ B_k образует абелеву подгруппу группы циклов $B_k \subset Z_k$. Очевидно, что B_k являются *образами* ($\text{im } \partial_{k+1}$) гомоморфизма ∂_{k+1} .

Конечно, произвольный цикл может и не быть границей. Такой цикл ничего не ограничивает и может быть «стянут» в точку. Поэтому говорят, что он *гомологичен нулю*: $z \sim 0$. Два цикла гомологичны друг другу $c_1 \sim c_2$, если они отличаются на некоторую границу²⁶ ∂h , т. е. $c_1 = c_2 + \partial h$ или $c_1 - c_2 \sim 0$. Это обстоятельство позволяет разбить группу Z_k на отдельные классы, каждый из которых содержит циклы, эквивалентные с точностью до ∂h . Например, параллели тора образуют один класс циклов, не являющихся границами. Другой класс образуют меридианы, пара которых ограничивает цилиндрическую поверхность тора между ними. Формально, вложенная последовательность $B_k \subseteq Z_k \subseteq C_k$ позволяет определить *фактор-группу k -мерных симплициальных гомологий*:

$$H_k = Z_k / B_k = \ker \partial_k / \text{im } \partial_{k+1}.$$

²⁵Т. е. отображение, сохраняющее бинарную операцию.

²⁶Примером может служить *кольцо*.

Число непересекающихся классов²⁷ H_k называют k -мерным числом Бетти β_k . Эти числа тесно связаны с *клеточным разбиением* топологических поверхностей. Например, тор можно рассматривать как объект, «склеенный» из более простых элементов — *клеток* различной размерности.

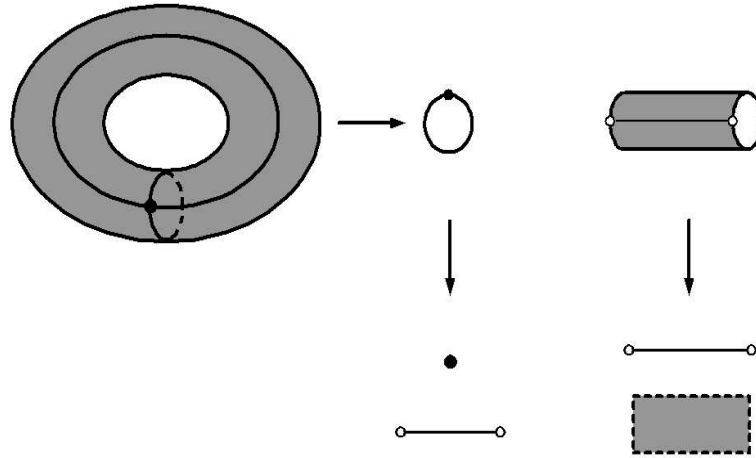


Рис. 23. Разложение тора на топологические клетки

Удалим из тора T^2 один из меридианов (рис. 23), т. е. 1-цикл. Края полученного разреза будем считать «отождествленными», т. е. склеенными. Удаление одной точки из вырезанного меридиана дает две клетки разной размерности: 0-клетку или точку, и отрезок или 1-клетку с отождествленными концами. Развернем то, что осталось от тора, в цилиндр и вырежем одну из образующих, т. е. 1-клетку. Это позволит развернуть цилиндр в прямоугольник или 2-клетку, у которого противоположные стороны попарно склеены. В результате мы получили все клетки, из которых составлен тор. Перечислим их по размерностям: одна 0-клетка, две 1-клетки и одна 2-клетка.

Рассмотрим теперь *гомологии* тора. Группа H_0 содержит точки поверхности T^2 : все циклы на торе, которые стягиваются в точку, и кратные им циклы. Поскольку все эти объекты гомологичны нулю $H_0(T^2) = \mathbb{Z}$ и, следовательно, $\beta_0 = 1$. Группа $H_1(T^2) = \mathbb{Z} \oplus \mathbb{Z}$ содержит два класса него-

²⁷Т. е. ранг группы.

мологичных (параллели и меридианы) 1-циклов, следовательно, $\beta_1 = 2$. Наконец, группа $H_2(T^2) = Z$. Она содержит только один класс 2-циклов, поверхность тора; поэтому $\beta_2 = 1$. Сравнивая полученные значения β_k с разложением тора на клетки, легко убедиться в их полной идентичности.

Числа Бетти связаны с характеристикой Эйлера поверхности формулой:

$$\chi = \sum_i (-1)^i \beta_i$$

так, что для T^2 мы получаем: $\chi = 1 - 2 + 1 = 0$. Для сферы S^2 число $\beta_0 = 1$ и, поскольку все 1-циклы сферы гомологичны нулю, $\beta_1 = 0$. Поверхность сферы дает $\beta_2 = 1$. Эти результаты полностью соответствуют разложению сферы на клетки. Таких клеток всего две: одна 0-мерная, выколота точка, и одна 2-х мерная, представляющая собой диск с отождествленной границей, который получается после растяжения проколота сферы на плоскость. Таким образом, $\chi(S^2) = 1 - 0 + 1 = 2$. Напомним, что согласно теореме Пуанкаре–Хопфа [42], эйлерова характеристика равна числу особенностей глобального векторного поля, заданного на поверхности. На торе такое поле не имеет особенностей и моделируется касательными к «обмотке» тора. Представим себе планету в форме тора. «Обмотка» тора — это ветровые потоки. Следуя им, воздушный шар, запущенный в любой точке тора, совершит «кругосветное путешествие» не пропустив ни одной точки поверхности планеты. Это и означает отсутствие особых точек ветрового векторного поля. Как известно, на сфере всегда существуют две сингулярности (два полюса), в которых невозможно определить направление касательных к S^2 векторов. Следовательно, воздушный шар никогда не попадет в эти точки. Таким образом, топологическое устройство многообразия определяет физику глобальных полей на нем. Я приведу один пример использования гомологий для упомянутых выше синоптических карт Солнечного магнитного поля.

Числа Бетти для $H - \alpha$ карт

Мы использовали для анализа синоптические $H - \alpha$ карты, показанные на рис. 20. Я напомним, что они представляют собой распределение фонового магнитного поля Солнца «с точностью до знака полярности» или, как еще говорят, распределение нейтральной линии, которая разделяет области с разной полярностью. Каждая карта соответствует одному обороту

Солнца. Выделим на карте области фиксированной полярности. Подсчитаем суммарное число циклов, окружающих каждую «дыру», т. е. область противоположной полярности. Полученная величина соответствует числу Бетти β_1 и в некотором смысле измеряет «пятнистость» дипольной компоненты магнитного поля. Временной ряд β_1 , полученный по 809 $H - \alpha$ картам, покрывающим интервал времени около 67 лет приведен на рис. 24. На том же рисунке показан график временного ряда чисел Вольфа, для

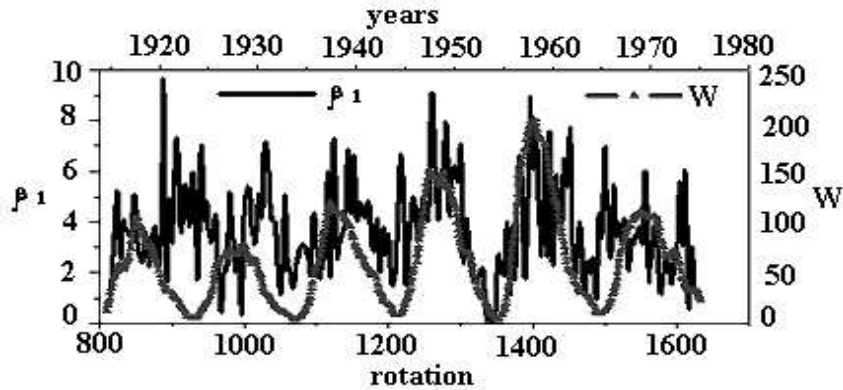


Рис. 24. Временной ряд чисел Бетти β_1 вместе с числами Вольфа для $H - \alpha$ карт

того же интервала времени. Оказалось, что спектр мощности полученных чисел Бетти содержит квазидвухлетнюю моду и периоды 11 и 22 года. Мы надеемся уточнить полученные результаты по полным синоптическим картам, используя так называемые «персистентные» числа Бетти, и получить значения β_2 . Они связаны с топологией «ручек», образованных 3D магнитными структурами глобального поля.

Путеводитель по литературе. Хорошее популярное введение в гомологии можно найти в очерке [60]. Гомологии и их применение к физике описаны в книге [61]. Академическое изложение гомологий и чисел Бетти содержит учебник [10]. Современный вариант теории гомологий с приложениями представлен монографией [62], доступной в электронном виде. Программное обеспечение по вычислениям кубических гомологий размещено на сайте [63]; его приложение к анализу распределенных систем описано в статье [65]. Альтернативный алгоритм для вычисления гомологий был предложен Тамалом Деєм [64]. Популярный обзор по вычислительной топологии [20] хорошо дополняется лекциями [66]. Заинтересованный читатель найдет много полезного в диссертациях Ванессы Робинс [22] и Афры Зомородиан [67]. В статьях [68, 69] рассмотрен индекс несвязности и его приложения к фракталам и динамике. Математическому введению в вычислительную топологию посвящены препринты [70–72]. Наконец, понятное введение в теорию Морса можно найти в статье Робина Формана [73].

Эпилог

Мы опять таки слышали только о
чем-то незаконченном, но разве у
нас не рождается надежда, что эти
знания должны будут продолжать
развиваться дальше и приведут к
новым открытиям, или нас самих
или других уже после нас? Да
разве мы сами не узнали уже
достаточно нового и
поразительного?

З. Фрейд
«Сновидения»

Основная задача любого эксперимента — извлечение явных знаний из массива данных. Если данные представлены в геометрической форме, численные характеристики можно получить либо метрическими (фрактальные размерности, ляпуновские показатели), либо геометрическими (функционалы Минковского) методами. Гораздо сложнее извлечь топологические свойства, которые более фундаментальны, чем упомянутые выше. Вы-

числительная топология исследует топологические свойства на объектах, измеренных или вычисленных с конечной точностью. Цифровая кривая, представленная набором пикселей, например, совсем не похожа на гладкую кривую, а цифровой контур, может потерять замкнутость при изменении увеличения. Иными словами, «цифровая» топология зависит от выбора решетки. Поэтому алгоритмы вычислительной топологии рассчитаны прежде всего на вычисление «персистентных» инвариантов, не зависящих от размера пикселей. Впрочем, изменение структуры паттерна с разрешением можно часто использовать как диагностический инструмент. Понятно, например, что число связанных кластеров на бинарном изображении зависит от разрешения. Подсчитаем это число для клеток с различными размерами. Оказывается, что скорость изменения этой величины с размером пикселя, которая называется индексом несвязности, не только различна для разных паттернов, но является чрезвычайно информативной характеристикой для фракталов и паттернов распределенной динамики. К сожалению, эти и многие другие интригующие вещи остались «за кадром». Увы, в лекции не нашлось места даже для основных морфологических операторов — «дилатации» и «эрозии», реализующих «арифметику Минковского». Не удалось рассказать и о Клеточных Нейронных сетях чрезвычайно удобных, для обработки изображения. Однако, в конце концов нельзя же пытаться затолкать слона в чайник! Разумней найти для него более достойное местоприемник. Возможно цитата из З. Фрейда, вынесенная в эпиграф, уменьшит скорбь Разочарованного Читателя.

Литература

1. Уолтер Г. Живой мозг. – М.: Мир, 1966. – 229 с.
2. Sauer T., Yorke J. A., Casdagli M. Embedology // *J. Statist. Phys.* – 1991. – v. 65. – pp. 579–616.
3. Parlitz U. and Merkwirth C. Prediction of spatiotemporal time series based on reconstructed local states // *Physical review letters* – 2000. – v. 84. – pp. 1890–1893.
4. Рабинович М. И., Фабрикант А. Л., Цимринг Л. Ш. Конечномерный пространственный беспорядок // *УФН* – 1992. – т. 42. – 42 с.
5. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. – М.: Мир, 1989. – 478 с.

6. Toussaint G., T. What is computational geometry? // *Proc. IEEE*. – 2001. – v. 80 – pp. 1347–1363.
URL: <http://cgm.cs.mcgill.ca/~godfried/publications.html>
7. Klette R. Digital geometry — the birth of a new discipline // *CITR-TR-79*. – 2001.
URL: <http://www.citr.auckland.ac.nz>
8. Calabi L., Hartnett W.E. Shape recognition, prairie fires, convex deficiencies and skeletons // *American Mathemat. Mon.* – 1968. – v. 75 – pp. 335–342.
9. Агаштейн М. Э., Микдал А. А. Как увидеть невидимое? // *Эксперимент на дисплее. Первые шаги вычислительной физики*. – М.: Наука, 1989. – с. 141–174.
10. Зейферт Г., Трельфаль В. Топология. – М.-Ижевск: РХД, 2001. – 448 с.
11. Weygaert Rien. van de Fragmenting the Universe. III. The construction and statistics of 3-D Voronoi tessellations // *Astrophysical J.* – 1994. – v. 283. – pp. 361–406.
12. Ikeuchi S., Turner E.L. Quasi-periodic structures in the large-scale galaxy distribution and three-dimensional Voronoi tessellation // *Mon. Not. R. Astr. Soc.*. – 1991. – v. 250. – pp. 519–522.
13. Gower J. C., Ross G. J. S. Minimum spanning trees and single linkage cluster analysis // *Appl. Statistics*. – 1969. – v. 18 – pp. 54–64.
14. Borucki W.J., Card D.H., and Lyle G. C. A method of using cluster analysis to study statistical dependence in multivariate data // *IEEE Transactions on computers*. – 1975. – v. 24 – No. 12. – pp. 1183–1191.
15. Bhavsar S. P. Nigelling E. Large-scale distribution of galaxies: Filamentary structure and visual bias // *Publ. Astron. Soc. Pacific*. – 1988. – v. 100. – pp. 1314–19.
16. Гренандер У. Лекции по теории образов. – т. 3. Регулярные структуры – М.: Мир, 1983. – 430 с.
17. Edelsbrunner H. The union of balls and its dual shape // *Discrete Comput. Geom.* – 1995. – v. 13. – pp. 415–440.
18. Edelsbrunner H., Kirkpatrick D. G., and Seidel R. On the shape of a set of points in the plane. // *IEEE Transactions on Information Theory*, IT-29(4) – 1983. – pp. 551–559.
19. Edelsbrunner H. and Mucke E. P. Simulation of Simplicity: a technique to cope with degenerate cases in geometric algorithms // *ACM Trans. Graphics*. – 1990. – v. 9. – pp. 66–104.
URL: <http://biogeometry.cs.duke.edu/software/alphashapes/>
20. Dey T.K., Edelsbrunner H. and Guha S. Computational Topology. – 1998.
URL: <http://www.cis.ohio-state.edu/~tamaldey/>

21. Edelsbrunner H. Spielereien mit Kreisen und Kugeln. Zum Thema Form und Verformung. Zur Kunst des Formalen Denkens. – pp. 153–171.
URL: <http://www.cs.duke.edu/~edels/>
22. Robins V. Computational Topology at Multiple Resolutions. – *PhD Thesis*. – 2000.
URL: <http://www.rspysse.anu.edu.au/~vbr110/thesis/thesis.html>
23. Edelsbrunner H. and Mucke E.P. Three-dimensional alpha shapes // *ACM Trans. Graphics* – 1994. – v. 13. – pp. 43–72.
URL: <http://biogeometry.cs.duke.edu/software/alphashapes/>
24. Edelsbrunner H. Biological applications of computational topology. Chapter 63 of Handbook of Discrete and Computational Geometry.
URL: <http://www.cs.duke.edu/~edels/>
25. Edelsbrunner H. Triangulations and meshes in computational geometry // *Acta Numerica*. – 2000. – pp. 133–213.
URL: <http://www.cs.duke.edu/~edels/>
26. Martinez T., Schulten K. A “Neural-Gas” network learns topologies // In Teuvo Kohonen, Kai Makisara, Olli Simula, Jari Kangas, editors, *Artificial Neural Networks*. Elsevier, Amsterdam. – 1991. – pp. 397–402.
URL: <http://www.ks.uiuc.edu/Publications/Papers>
27. Martinez T., Schulten K. Topology representing networks // *Neural Networks*. – 1994. – v. 7. – pp. 507–22.
URL: <http://www.ks.uiuc.edu/Publications/Papers>
28. Fritzke B. Growing cell structures – a self-organizing network for unsupervised and supervised learning // *ICSI TR 93-026*
URL: <http://www.ki.inf.tu-dresden.de/~fritzke/>
29. Кендалл М., Моран П. Геометрические вероятности. – М.: Наука, 1972. – 192 с.
30. Секей Г. Парадоксы в теории вероятностей и математической статистике. – М.: Мир, 1990. – 240 с.
31. Новиков А. Распознавание образов с помощью интегральной геометрии // В сб. *Принципы самоорганизации*. – М.: Мир, 1966. – с. 428–452.
32. Мильчина В.
URL: <http://vivovoco.nns.ru/VV/PAPERS/LITRA/BUFMILCH.HTM>
33. Gridgeman N. T. Geometric probability and the number π // *Scripta Mathematica*. – 1960. – v. 25 – pp. 183–195.
34. Сантало Л. Интегральная геометрия и геометрические вероятности. – М.: Наука, 1983. – 358 с.
35. Сантало Л. А. Введение в интегральную геометрию. – М.: Издательство иностранной литературы, 1956. – 183 с.

36. Бляшке В. Введение в дифференциальную геометрию. – М.: Государственное издательство технико-теоретической литературы, 1957. – 223 с.
37. Flanders H. Differential forms with applications to the physical sciences. – New York, London: Academic press, 1963. – 203 с.
38. Serra J. Image analysis and mathematical morphology. – Academic Press, Inc. – 1982. – 603 p.
39. Michielsen K., De Raedt H., De Hosson J. Th. M. Aspects of mathematical morphology // *Advances in imaging and electron physics* – 2002. – v. 125. – pp. 119–194.
URL: <http://rugth30.phys.rug.nl/compphys0/publicat.htm>
40. Michielsen K. and De Raedt H. Integral-Geometry Morphological Image Analysis // *Phys. Rep.* – 2001. – v. 347. – pp. 461–538.
URL: <http://rugth30.phys.rug.nl/compphys0/publicat.htm>
41. Mecke K. R., Wagner H. Euler Characteristic and Related measures for random geometric sets // *J. Statistical Phys.* – 1991. – v. 64 – pp. 843–849.
42. Chillingworth D. Differential topology with a view to applications. – Pitman Press, 1976. – 291 pp.
43. Минский М., Пейперт С. Перцептроны. – М.: Мир, 1971. – 261 с.
44. Матерон Ж. Случайные множества и интегральная геометрия. – М.: Мир, 1978. – 318 с.
45. Свешников А. А. Прикладные методы теории случайных функций. – М.: Наука, 1968. – 464 с.
46. Лонге-Хиггинс М. С. Статистический анализ случайной движущей поверхности // *Ветровые волны*. – М.: ИЛ, 1962. – с. 125–218 – № 3. – с. 94–99.
47. Adler R. J. The geometry of random fields. – New York: Wiley, 1981. – 280 с.
48. Adler R. J., Taylor J. E. Random fields and their geometry
URL: <http://iew3.technion.ac.il/~radler/publications.html>
49. Worsley K. J. The geometry of random fields // *Chance*. – 1996. – v. 9. – pp. 27–40.
URL: <http://www.math.mcgill.ca/~keith/>
50. Worsley K. J. Estimating the number of peaks in a random field using the Hadwiger characteristic of excursion sets, with applications to medical images // *Annals of Statistics*. – 1995. – v. 23. – pp. 640–669.
URL: <http://www.math.mcgill.ca/~keith/>
51. Worsley K. J. Local maxima and the expected Euler characteristic of excursion sets of χ^2 , F and t fields // *Advances in Applied Probability*. – 1994. – v. 26. – pp. 13–42.
URL: <http://www.math.mcgill.ca/~keith/>

52. Makarenko N., Karimova L., Terechov A., Novak M. M. Topological classification of radioactive contamination // *Physica A*. – 2001. – v. 289. – pp. 278–289.
53. Karimova L., Makarenko N. Diagnosis of stochastic fields by the mathematical morphology and computational topology methods // *Nuclear instruments and methods in physics research A*. – 2003. – v. 502. – pp. 802–804.
54. Макаренко Н. Г., Каримова Л. М., Терехов А. Г., Кардашев А. В. Функционалы Минковского и сравнение дискретных выборок в сейсмологии // *Физика Земли*. – 2000. – № 4. – pp. 48–52.
55. Mordvinov A. V., Salakhutdinova I. I., Plyusnina L. A., Makarenko N. G., and Karimova L. M. The topology of background magnetic fields and Solar flare activity // *Solar physics*. – 2002. – v. 211. – pp. 241–253.
56. Bardeen J. M., Bond J. R., Kaiser N., Szalay A. S. The statistics of peaks of Gaussian random fields // *Astrophysical J.*. – 1986. – v. 304. – pp. 15–61.
57. Hamilton A. J. S. The Topology of fractal universes // *Publication of the Astronomical Soc. of the Pacific*. – 1988. – v. 100. – pp. 1343–50.
58. Ryden B. S., Melott A. L., Craig D. A., et al. The area of isodensity contours in cosmological models and Galaxy surveys // *Astrophysical J.* – 1989. – v. 340. – pp. 647–660.
59. Adams F. C. A topological/geometrical approach to the study of astrophysical maps. // *Astrophysical J.* – 1992. – v. 387. – pp. 572–90.
60. Болтянский В. Г., Ефремович В. А. Очерк основных идей топологии. – М.: Гос. издательство технико-теоретической литературы, 1957–1960. – вып. 2–6 – 343 с.
61. Шаниро И. С., Ольшанецкий М. А. Лекции по топологии для физиков. – Москва-Ижевск: РХД, 2001. – 128 с.
62. T. Kaczynski T., Mischaikow K., Mrozek M. Computing Homology // *Homology, Homotopy and Applications*. – 2002. – v. 5. – pp. 233–256.
URL: <http://www.math.gatech.edu/~mischaik/papers/paperlist.html>
63. Computational Homology Program
URL: <http://www.math.gatech.edu/%7Echom/>
64. T. K. Dey and S. Guha. Computing homology groups of simplicial complexes in R^3 // *Journal of ACM*. – 1998. – v. 45 – No. 2 – pp. 266–287.
URL: <http://www.cis.ohio-state.edu/~tamaidey/>
65. Mischaikow K., Gameire W., Kalies W. Topological Characterization of Spatial Temporal Chaos
URL: <http://www.math.gatech.edu/~mischaik/papers/paperlist.html>
66. Zomorodian A. Introduction to Computational Topology.
URL: <http://graphics.stanford.edu/courses/cs468-02-fall/schedule.html>

- 67. Zomorodian A. Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes. *PhD Thesis*. – 2001.
URL: <http://http://www.graphics.stanford.edu/~afra/>
- 68. Robins V., Meiss J. D., and Bradley E. Computing connectedness: Disconnectedness and discreteness // *Physica D* – 2000. – v. 139. – pp. 276–300.
URL: <http://wwwrphysse.anu.edu.au/~vbr110/vbr.html>
- 69. Robins V., Meiss J. D., and Bradley E. Computing connectedness: An exercise in computational topology // *Nonlinearity* 11 – 1998. – pp. 913–922.
URL: <http://wwwrphysse.anu.edu.au/~vbr110/vbr.html>
- 70. Klette R. Digital topology for image analysis. PART I – basics and planar image carriers // *CITR–TR–101*. – 2001.
URL: <http://www.citr.auckland.ac.nz>
- 71. Eckhardt U., Latecki L. Digital topology. – *Preprint*. – 1999. – 22 pp.
URL: <http://cosmic.rrz.uni-hamburg.de/docs/wcat/math.html>
- 72. Kovalevsky V. Algorithms and data structures for computer topology. – In *Bertrand G., Imiya A., Klette R. (Eds.) Digital and Image Geometry LNCS 2243*. – Springer, 2001. – pp. 37–58.
URL: <http://www.Kovalevsky.dc/PublicationsE.htm>
- 73. Forman R. A user's guide to discrete Morse theory
URL: <http://math.rice.edu/~forman/>

Николай Григорьевич МАКАРЕНКО, ведущий научный сотрудник, кандидат физико-математических наук, руководитель группы в Лаборатории компьютерного моделирования (Институт математики, Алма-Ата, Казахстан). Области научных интересов: фрактальная геометрия, вычислительная топология, алгоритмическое моделирование, детерминированный хаос, нейронные сети, физика Солнца. Имеет более 50 научных публикаций.

НАУЧНАЯ СЕССИЯ МИФИ–2004

НЕЙРОИНФОРМАТИКА–2004

VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ
Часть 2

Оригинал-макет подготовлен Ю. В. Тюменцевым
с использованием издательского пакета L^AT_EX 2_ε
и набора PostScript–шрифтов PSCyr

Подписано в печать 25.11.2003 г. Формат 60 × 84 1/16

Печ. л. 12, 5. Тираж 200 экз. Заказ №

*Московский инженерно-физический институт
(государственный университет)
Типография МИФИ
115409, Москва, Каширское шоссе, 31*