

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЗАДАЧАХ УПРАВЛЕНИЯ

УДК 658.012.011.56

© 1994 г. В. Н. ЗАХАРОВ, С. В. УЛЬЯНОВ

НЕЧЕТКИЕ МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОМЫШЛЕННЫХ РЕГУЛЯТОРОВ И СИСТЕМ УПРАВЛЕНИЯ. IV. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Обсуждаются проблемы имитационного моделирования промышленных регуляторов и систем управления с различной степенью интеллектуальности. Подчеркивается, что имитационное моделирование является неотъемлемым этапом проектирования управляющих систем такого класса. Рассматриваются основные принципы проектирования баз знаний интеллектуальных систем управления методами имитационного моделирования. Описываются базовые механизмы обработки больших массивов информации и поиска решений в слабоструктурированных средах в виде генетических алгоритмов и нейронных сетей, составляющие основу инструментария для формирования баз знаний. Предлагаемый подход позволяет установить объективную форму проверки экспертных знаний и организовать эффективный поиск по всей области возможных решений. Приведены примеры, иллюстрирующие работу описываемых механизмов.

Введение. Современные методы построения нечетких моделей регуляторов и систем управления получили широкое распространение в процессах проектирования и разработки промышленных интеллектуальных управляющих систем [1—32]. Положительный опыт использования моделей нечетких регуляторов и систем управления [33—35] заложил основу развития новой инвариантной информационной технологии — технологии построения структур интеллектуальных нечетких контроллеров, использующих принцип самоорганизации [36, 37] или, в более общем случае, активные гибридные нечеткие экспертные системы [38, 39]. При этом нечеткие регуляторы типа [40—44] (с уровнем интеллектуальности «в малом» по классификации [45]) используют в качестве базы знаний так называемые ТЛП — таблицы лингвистических правил (lookup-table) с различными уровнями истинности приближенных суждений и семантической выразительности продукционных правил. Аппаратная реализация таких ТЛП [46] для конкретных моделей логического вывода осуществляется на основе нечетких процессоров с различной архитектурой [47, 48].

Разнообразие моделей нечеткого логического вывода привело к необходимости развития методов формулирования структур самоорганизующихся нечетких систем управления [49, 50] за счет повышения уровня интеллектуальности [51—53] или развития базы знаний с компиляцией различных ТЛП на языках высокого уровня [54]. По мнению [55] компромисс между точностью и сложностью аппаратной реализации может быть достигнут за счет развития моделей нечетких процессоров в виде построения так называемых частных интеллектуальных машин типа MIQ (Machine Intelligence Quotient).

Согласно разработанным в [56, 57] принципам проектирования, степень возникаемой неопределенности (в виде энтропии) и формируемая необходимая база знаний (БЗ) определяют в данном случае требуемый уровень интеллектуальности разрабатываемой системы управления с последующей оценкой достаточных средств аппаратной поддержки разрабатываемой интеллектуальной САУ.

Теоретические и практические результаты разработок нечетких систем управления привели к выводу о перспективности использования так называемых генетических алгоритмов (ГА) и нечетких нейронных сетей (ННС) при моделировании проектируемых БЗ и структур интеллектуальных систем управления [58—68]. В то же время анализ использования интеллектуальных нечетких САУ (ИНСАУ) в проблемно-ориентированных задачах теории управления показал предельные возможности использования отдельно нейронных сетей и нечетких систем [58]. Развиваемая в настоящее время новая инвариантная информационная технология проектирования систем управления сложными объектами [17, 18, 36—38, 58—60, 64, 69] опирается на теорию нечетких систем как метод улучшения динамических характеристик управляемого объекта [39, 70] и на теорию ННС как метод формирования БЗ в ИНСАУ. Как будет показано далее, снятие неопределенности в процессах проектирования ИНСАУ осуществляется методами имитационного моделирования на базе ГА [71] и ННС [58, 64] с последующим использованием сформированных БЗ в активных экспертных системах (АЭС) нечетких регуляторов [39]. Включение в контур управления АЭС позволяет существенным образом улучшить динамику управляемого объекта и расширить его функциональные возможности за счет применения интеллектуальных средств объективно сформулированной методикой имитационного моделирования БЗ.

Совместное использование модели ГА и нейронных сетей (НС) позволяет устранить основной недостаток формируемых БЗ АЭС — субъективность и противоречивость знаний экспертов, а также неполное знание всех решений в заданной предметной области. Главная особенность рассматриваемого подхода состоит в использовании возможностей ГА находить субоптимальные (пригодные) решения на всем множестве возможных решений и возможностей НС проверять ограниченное число найденных субоптимальных решений с целью отбора оптимального (самого пригодного) решения. Это позволяет избежать излишних вычислительных затрат комбинаторного взрыва возможных вариантов с ростом размерности задачи, а также дает возможность проверить каждое оригинальное решение, предлагаемое экспертом.

Предложенная в [39, 56, 57] методология проектирования интеллектуальных систем управления (FZUP-методология) опирается именно на такой подход к имитационному моделированию с целью формирования смыслового содержания БЗ и согласования уровней выбираемых исполнительных устройств и уровней интеллектуальности.

Настоящая статья, являющаяся продолжением [39], посвящена изложению основных принципов построения и особенностей использования моделей ГА и НС в имитационном моделировании при проектировании БЗ для ИНСАУ.

1. Задачи имитационного моделирования при разработке интеллектуальных САУ. Анализ традиционных подходов к проектированию регуляторов для реальных объектов управления, функционирующих в условиях широкого изменения параметров структуры или хаотически организованной среде, показал [72—76] необходимость и целесообразность включения в структуру управления интеллектуальных средств при выборе вида и формировании замкнутых обратных связей. Это особенно важно при создании робастных и адаптивных регуляторов для неустойчивых объектов управления [77, 78], функционирующих в неопределенных средах [74]. В этом случае приходится прибегать к системам управления, основанным на знаниях [79—85], в частности к нечетким системам управления [86, 87], сочетающим в себе свойства робастности, адаптации и самоорганизации [36, 37, 49—52].

Рассмотрим особенности решения главной задачи имитационного моделирования интеллектуальных САУ — формирование требуемой БЗ в виде ТЛП — на примере комбинированного нечеткого регулятора, описанного в [56].

Длина алгоритмического описания лингвистической аппроксимации объектов и процессов управления в виде продукционных правил определяет точность аппроксимации и степень адекватности модели реальному объекту. Поэтому часто стремятся увеличить количество продукционных правил (особенно при разработке структур нечетких регуляторов экспертными методами). Увеличение количества продукционных правил нечеткого логического вывода в ТЛП для управления сложными процессами приводит к формированию линейной зависимости выходного сигнала блока дефазификации нечеткого контроллера от входного сигнала. Этот факт означает, что при достаточно большом числе продукционных правил структура нечеткого контроллера может быть аппроксимирована с достаточно большой точностью структурой ПИД-регулятора [88]. Но и структура ПИД-регулятора может быть поставлена в соответствие структуре нечеткого контроллера [89, 90]. Поэтому при проектировании структур нечетких контроллеров разработчики вынуждены снижать количество продукционных правил в ТЛП за счет увеличения их семантической выразительности и адекватности описываемым процессам управления и усовершенствовать алгоритмические процедуры вывода с учетом функций, реализуемых дефазификатором [91, 92]. Снижение аппаратной сложности нечетких процессоров за счет декомпозиции ТЛП [93—95] приводит к релукции данных и повышению степени риска появления в ней «саботажных» продукционных правил. Устранение таких явлений осуществляется за счет совершенствования иерархической структуры нечетких алгоритмов управления [96], приводящего к снижению числа используемых процедур адаптации с применением моделей-эталонов [97], а также за счет реализации более гибких методов нечеткого логического вывода [98].

Решение задачи управления сложными объектами в условиях неопределенности возможно также на основе использования комбинированного принципа проектирования, сочетающего в себе преимущества работы традиционных регуляторов (при известной структуре модели-эталона объекта управления) и нечувствительность (грубость) нечетких регуляторов к изменению параметров объекта или внешней среды [99—109].

В качестве примера можно привести задачу поиска необходимого управления для перевода маятника из положения устойчивого равновесия в неустойчивое (перевернутое) положение и/или удержание его в устойчивом перевернутом состоянии [39, 103, 104]. При этом с помощью декомпозиции общая задача разбивается на подзадачи типа управления движением каретки и установленным на ней маятником на основе традиционных методов П-, И-регуляторов и их комбинаций [105]; привлечением более сложных скользящих режимов и алгоритмов типа систем с переменной структурой [106] и их нечетких аналогов [107]; использованием только интеллектуальных средств [108—111] и их нечетких аналогов [112].

Анализ полученных решений перечисленных подзадач показал, что при наличии известной (достаточно простой) модели объекта управления и несложных ограничениях на процессы управления традиционные регуляторы могут выполнять задачу управления в отдельных случаях лучше нечетких контроллеров (например, при ограниченном числе продукционных правил [113]). Но с усложнением моделей объектов управления, введением существенных ограничений на управление и увеличением степени неопределенности нечеткие регуляторы и системы управления в целом функционируют эффективнее [70].

Разработка нечеткого регулятора (системы управления, интеллектуальной в малом) и, тем более, управляющих систем с более высокими уровнями интеллектуальности неразрывно связана с необходимостью организации соответствующей (в данном случае, нечеткой) БЗ. Проблема организации требуемой БЗ (например, в виде ТЛП) является в достаточной мере сложной, поскольку

содержимое БЗ, а также механизмы хранения, пополнения и обработки знаний (в частности, нечетких правил) оказывают существенное влияние на динамические характеристики проектируемой системы. Эта задача не может быть решена без тщательного анализа функционирования системы путем имитационного моделирования с применением двух важнейших типов моделей: генетических алгоритмов [114—117] и нечетких нейронных сетей [58—69, 118, 119]. Модели первого типа позволяют на множестве возможных решений найти наиболее близкие к искомому решению (множество субоптимальных решений). Модели второго типа дают возможность по результатам моделирования сформировать окончательное множество оптимальных решений.

Имитационное моделирование является неотъемлемым этапом эскизного проектирования интеллектуальной САУ. С его помощью оказывается возможным установить необходимое соответствие функциональных особенностей исполнительного уровня требуемому уровню интеллектуальности проектируемой САУ и осуществить в случае необходимости перераспределение функций этих уровней с учетом предельных возможностей аппаратных средств поддержки используемых механизмов [39, 56—59].

Метод лингвистической аппроксимации сложных объектов и процессов управления, функционирующих в условиях неопределенности априорной информации рассматривается в данном случае как один из возможных подходов к решению проблемы понижения сложности описания динамического поведения физических систем (как конечных объектов) на логическом и алгоритмическом уровнях [70, 120, 121]. Дальнейшее развитие данного метода привело к необходимости доказательства правомерности использования нечетких систем как универсальных аппроксиматоров структур сложных систем управления [122, 123] на основе заданного класса базисных функций принадлежности [113, 124]. Включение в нейронные сети, осуществляющие имитационное моделирование, элементов теории нечетких систем позволяет реализовать редукцию описания процессов функционирования интеллектуальных систем управления к изменению условий работы объектов управления [50, 58].

Редукция описания процессов функционирования интеллектуальных САУ сложными промышленными объектами [125] на основе теории нечетких множеств в виде лингвистической аппроксимации приводит к совершенствованию методологии имитационного моделирования процессов управления с использованием качественных моделей приближенных суждений [126], с проверкой адекватности разрабатываемых моделей [70, 127].

Одновременно введение понятия степени истинности приближенных суждений об адекватности построенной имитационной модели процессов управления реальным объектом приводит к необходимости введения понятия систем управления с различными уровнями интеллектуальности. Такой подход и дает возможность создания методологии проектирования на основе введенных в [39, 56, 57] принципов с использованием методов имитационного моделирования.

Обобщенная структурная модель нечеткого регулятора, БЗ которого формируется в виде ТЛП и корректируется методами имитационного моделирования, изображена на рис. 1. Данная модель содержит уровни адаптации, самоорганизации, обучения, позволяющие в зависимости от проблемной ориентации реализовать различные уровни интеллектуальности нечеткого регулятора. Рассмотрим далее особенности методологии имитационного моделирования БЗ для регулятора типа, изображенного на рис. 1.

2. Методология имитационного моделирования интеллектуальных САУ. Анализ физических аспектов теории нечетких интеллектуальных систем как универсальных аппроксиматоров показал [56, 128], что формирование БЗ в таких системах осуществляется на основе принципа описания динамического поведения систем управления как неравновесных открытых (в информационно-термодинамическом смысле) систем, в которых осуществляется обмен информацией с внешней средой. Такой подход позволил сформулировать принципы проектиро-

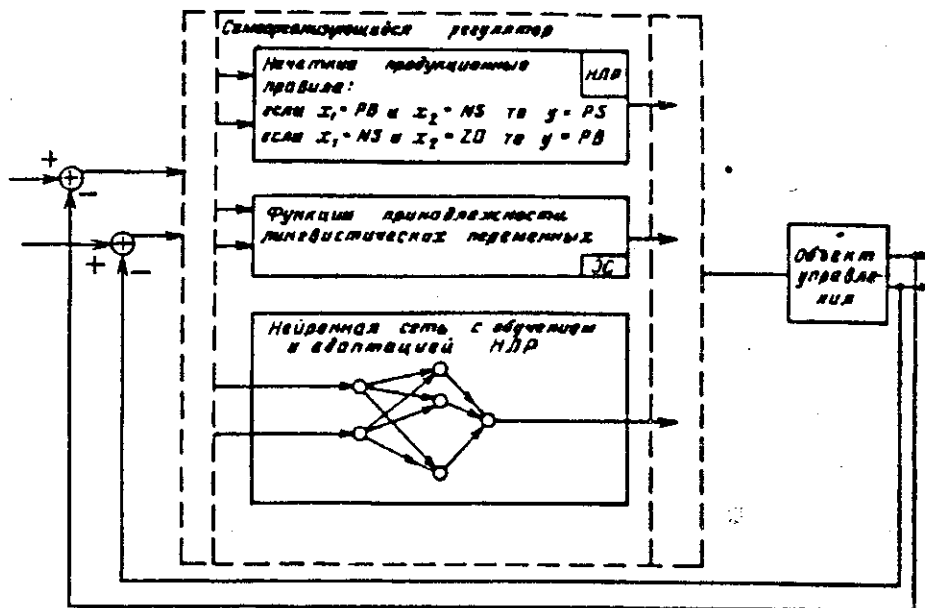


Рис. 1. Обобщенная структурная модель нечеткого интеллектуального регулятора

вания интеллектуальных САУ лировать пять принципов проектирования интеллектуальных систем управления с различными уровнями интеллектуальности [56, 57].

Введение алгоритмов приближенных суждений для процедур принятия решений в блоке логического вывода в нечетких системах управления и регуляторах привело, в свою очередь, к необходимости исследования и снятия противоречий в выводах на базе немонотонных логик, содержащих понятия несовместимых правил вывода в БЗ. При этом сами БЗ формируются объективно на основе редукций отношений «вход — выход» процедурами дедуктивных приближенных суждений [130, 131].

Одной из возможных логических основ разрешения подобного рода противоречий призвана служить квантовая релятивистская логика [132], позволяющая формировать цепочки непротиворечивых суждений в виде диалоговых процедур типа «вопрос — ответ» о состояниях исследуемой системы. В основе данных процедур лежит анализ состояний системы по соотношениям вида: «движение системы порождает логическую форму»; «логическая форма порождает движение системы» [133]. Примером может служить формирование фазового портрета динамической системы по фазовым координатам движения и изменения вида фазового портрета (например, образование скользящих режимов), приводящие к новым видам движения системы. В первом случае фазовый портрет является формой, образованной движением системы; во втором случае изменение формы (фазового портрета) порождает новые виды движения системы. В общем случае движение системы порождает устойчивые логические коагеративные структуры как результат действия синергетических процессов [135]. Анализ сформированных логических структур проводится на основе активных диалоговых процедур квантовой логики [136—138]. В этом случае осуществляется имитационное моделирование соотношений типа ответов «да — нет» на вопросы о состояниях динамической системы [136] с различными степенями истинности суждений на открытом интервале (0,1) [139].

З а м е ч а н и е 1. Данная процедура имитационного моделирования обоснована тем, что описанию функционирования реального объекта присуща некая объективная мера нечеткости [140]. Исследования полноты соответствующей оценки

адекватности описания функционирования объекта управления показали [139], что значения истинности могут находиться в открытом интервале (0,1). Данный результат означает, что в общем случае для класса динамических систем истинность суждений об адекватности составляющих подсистем своим моделям не может принципиально принимать булевы значения {0,1}. Оказалось, что такой класс систем описывается в рамках квантовой логики [136—138, 140—143], которая оперирует с характеристическими функциями свойств объектов управления на глубинных семантических уровнях в отличие от других логик синтаксического анализа [144, с. 251]. В частности, корректное описание логических связок квантовой логики требует более широкого привлечения семантики по сравнению с соответствующими связками классической логики синтаксического уровня [138, с. 220]. С помощью имитационного моделирования проверяется сохранение основных свойств качественного описания исследуемой системы и эффективность алгоритма приближенных суждений.

Как уже отмечалось, в методологии имитационного моделирования БЗ нечетких регуляторов и систем управления и механизмов вывода на знаниях в качестве инструментария широко используются модели генетических алгоритмов (ГА) и нейронных сетей (НС). Как те, так и другие модели являются разновидностями аналогичных представлений биологических процессов, но используют различные подходы к собственной структурной организации.

В основу построения модели первого типа положена аналогия с известным биологическим механизмом — принципом отбора по Дарвину. ГА позволяет на множестве возможных решений осуществлять локализацию относительно наиболее подходящих решений для данной проблемной области. Это становится возможным за счет организации специальных процедур проверки и отбрасывания непригодных решений. Сгруппированные решения передаются в качестве входной информации на НС.

Нейронная сеть использует в качестве биологической аналогии принцип работы связанных нейронов, в сети из которых осуществляется преобразование информации, а за счет перераспределения весов находятся искомые решения. С помощью НС осуществляется проверка предлагаемых решений и выбор оптимального из них. Таким образом, представленная на рис. 1 структура самоорганизующегося регулятора использует для формирования базы знаний в качестве инструментария методы имитационного моделирования с проверкой (выбором) оптимального решения. В качестве такого решения могут быть параметры функций принадлежности или продукционные правила в ТЛП.

Прежде чем переходить к особенностям применения НС и ГА в качестве основы инструментария в задачах имитационного моделирования нечетких контроллеров и интеллектуальных нечетких САУ, напомним основные положения и определения теории НС, а затем рассмотрим их нечеткие аналоги, то есть наиболее распространенные модели в исследованиях в области прикладных нейронных систем.

3. Искусственные нейронные сети. В основу искусственных НС [60, 61, 147—156] положено представление об изолированном биологическом нейроне, изображенном схематично на рис. 2, а, и взаимодействующим с другими нейронами. Современное представление утверждает, что биологический нейрон осуществляет две различные математические операции, распределенные между синапсом (объединяющей точкой между аксоном и дендритом) и сомой (телом клетки нейрона). Указанные математические операции, совершаемые нейроном, часто называют синаптической операцией и соматической операцией. С биологической точки зрения указанные операции различны и имеют свою физическую природу. При моделировании эти операции объединяют (например, параметр смещения в виде оценки истинности приписывают синаптической операции). Модели имитационного моделирования на искусственных нейронных сетях построены по аналогии с моделями работы естественного мозга, включающими взаимосвязанную сеть биологических нейронов количеством от 10^{10} до 10^{12} нейронов

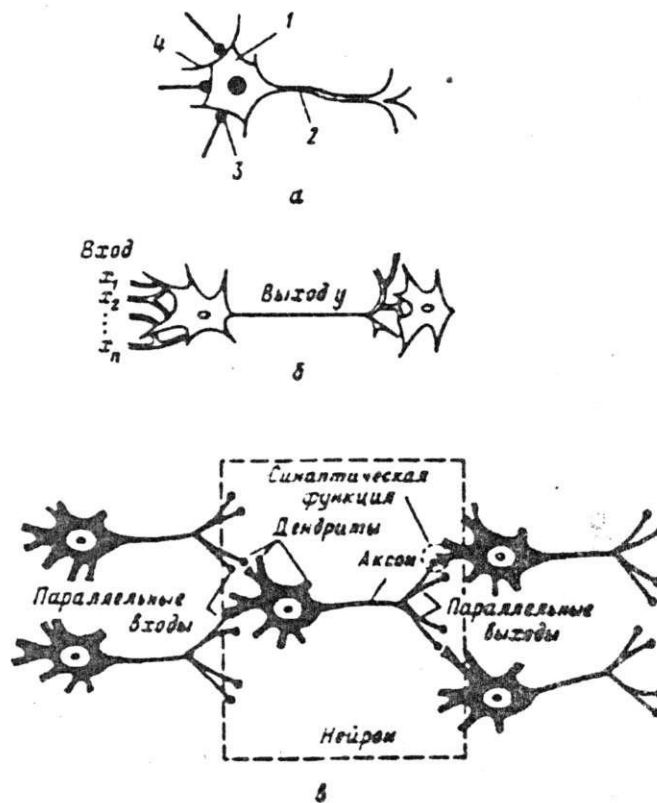


Рис. 2. Модели нейрона: а — биологический аналог изолированного нейрона (1 — тело клетки, 2 — аксон, 3 — синаптическая связь, 4 — дендрит); б, в — биологические аналоги статически (б) и динамически (в) взаимодействующих нейронов

[157]. Современные супер-ЭВМ содержат до 10^6 транзисторов, каждый из которых имеет связь от 2 до 4 транзисторов [158]. В естественном мозге, содержащем 10^{10} нейронов, каждый нейрон имеет связи приблизительно с 10^4 другими нейронами. Электрические сигналы поступают в виде суммарного сигнала через синаптические связи в тело клетки (сому), ее возбуждают (нелинейное преобразование сигналов), и через аксон преобразованный сигнал передается в суммарном виде на другой нейрон. На рис. 2, б приведена схема биологического аналога двух статически взаимодействующих нейронов, а на рис. 2, в — схема динамически взаимодействующих нейронов [159]. Формальное представление модели нейрона, отражающее основные свойства биологического нейрона (см. рис. 2, а), показано на рис. 3, а. На вход нейрона поступают n входных сигналов x_i ($i = 1, \dots, n$), а выходом нейрона является сигнал y ; внутреннее состояние нейрона определяется через взвешенную сумму входных сигналов (синаптическая операция), а выходной сигнал нейрона y определяется через сигмоидальную функцию f (соматическая операция) в виде

$$y(t) = f\left(\sum_{i=1}^n w_i x_i(t) - \theta\right), \quad (3.1)$$

где $\sum_{i=1}^n w_i x_i(t) = I(t)$ — входной суммарный сигнал, w_i — весовые коэффициенты, θ — смещение модуля нейрона (параметр правдоподобия), t — время, n — количество входов. Физически весовые коэффициенты w_i характеризуют силу синап-

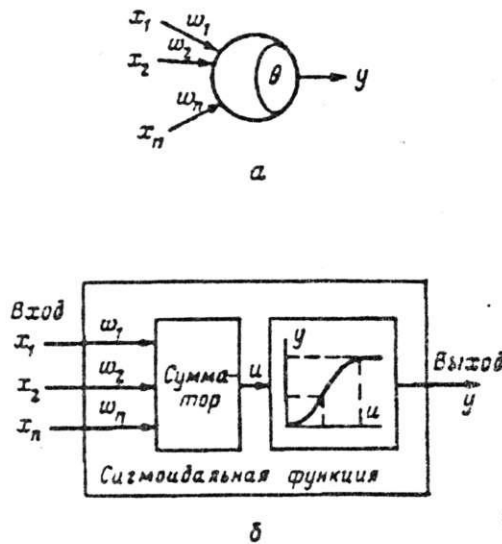


Рис. 3. Представления формального нейрона: а — математическая модель; б — структурная схема

тических связей нейрона. Наиболее применяемыми сигмоидальными функциями в (3.1) являются функции вида

$$f(x) = (1 + e^{-ax})^{-1} (0 < f(x) < 1), \quad ax = (I - \theta), \quad (3.2)$$

$$f(x) = \frac{1 - \exp(-ax)}{1 + \exp(-ax)} \quad (-1 < f(x) < 1). \quad (3.3)$$

На рис. 3, б схематично представлено описание преобразования сигнала нейроном с сигмоидальной функцией вида (3.2).

В общем случае динамический модуль нейрона (см. рис. 2, в) включает в себя элементы задержки, синаптические весовые коэффициенты прямых и обратных связей и нелинейный оператор преобразования входного суммарного сигнала [159]. Допустим, что имеются два элемента задержки (z^{-1}), два синаптических весовых коэффициента прямых a_{ff} (ff — feedforward) и обратных b_{fb} (fb — feedback) связей, $a_{ff} = [a_0, a_1, a_2]$, $b_{fb} = [b_1, b_2]$. Выходной суммарный сигнал $V_1(k)$ (k — дискретный индекс параметра времени) для блока синаптической связи будет описываться дискретным уравнением вида

$$V_1(k) = -b_1 V_1(k-1) - b_2 V_1(k-2) + a_0 x(k) + a_1 x(k-1) + a_2 x(k-2). \quad (3.4)$$

Здесь $x(k) \in R^n$ и описывает входной сигнал нейрона; $V_1(k) \in R^1$ является выходным сигналом динамического модуля синаптических связей. Для уравнения (3.4) передаточная функция имеет вид:

$$W(k, a_{ff}, b_{fb}) = \frac{V_1(k)}{x(k)} = \frac{[a_0 + a_1 z^{-1} + a_2 z^{-2}]}{[1 + b_1 z^{-1} + b_2 z^{-2}]},$$

где z^{-1} — элемент задержки. Нелинейный оператор отображения сигнала $V_1(k)$ в сигнал выхода нейрона (соматическая операция) описывается в виде $u(k) = f^{\text{sig}}[g_s, V_1(k)]$, где функция f описывается одним из операторов (3.2) или (3.3).

В литературе по искусственным нейронным сетям часто используются следующие обозначения: для синаптической операции входного суммарного сигнала на нейрон

$$\text{net}_j = \sum_{i=1}^n x_i w_{ji}, \quad w_{ji} = (w_{j1}, \dots, w_{jn})$$

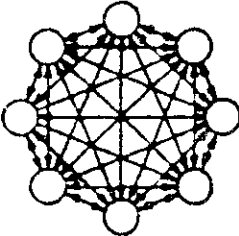
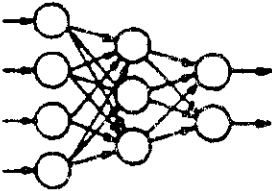
Вид моделей	Модель Хопфилда	Модель Румелхарта
Область применения	Оптимизация связей	Классификация, анализ и интерполяция
Вид структуры нейронной сети		
Отношение вход-выход	Входы и выходы нейронов не разделены (интегрированная связь)	Иерархическая связь между уровнями входов и выходов
Описание нейрона	С внутренними связями	Без внутренних связей
Тип поведения сети	Динамический	Статический
Характеристика поведения сети	Независимое поведение (минимизация энергии)	Отражение состояний входов на составные выходы
Тип обучения	По минимуму энергии	По отклонению моделируемых характеристик от заданных

Рис. 4. Принципы построения и основные характеристики известных моделей нейронных сетей

и выходного сигнала нейрона (соматической операции)

$$\text{out}_i = \begin{cases} 1, & \text{если } (\text{net}_i - \theta) \geq 0; -1 \text{ (или } 0), & \text{если } (\text{net}_i - \theta) < 0; \end{cases} \quad (3.5)$$

$$\text{out}_i = \max(b, \min(0, (\text{net}_i - \theta))) - \text{расстояние Хэмминга} \quad (3.6)$$

$$\text{out}_i = (1 + \exp(-(\text{net}_i - \theta)))^{-1} - \text{сигмоидальная функция.} \quad (3.7)$$

В зависимости от принципа взаимодействия выходного сигнала out_i изолированного нейрона с другими нейронами в сети определяется тип искусственной нейронной сети. Так, например, выходной сигнал отдельно взятого нейрона может быть связан со входами нейронов, расположенных на глубоком уровне сети; с рядом постоянно расположенными нейронами: с нейронами, позиционно зависящими от времени; с обратными связями нейронов, расположенных на одном внутреннем слое, с упорядоченными по приоритетам внутренним связям нейронов и т. п. В теории искусственных нейронных сетей наиболее распространенными моделями являются модели Хопфилда, Хэмминга, модель адаптивной двунаправленной ассоциативной памяти АВАН (Adaptive Bidirectional Associative Memory) модель Кохонена (с обучением) и модель обратного распространения ошибки (back-propagation of errors). Следует отметить, что модели Хопфилда и Хэмминга оперируют только бинарными образами, в то время как модели АВАН, Кохонена и обратного распространения ошибки могут оперировать обоими типами (бинарными и действительными).

В таблице на рис. 4 приведены два наиболее распространенных вида моделей Хопфилда и Румелхарта (обратное распространение ошибки) с кратким описанием их отличий и областей применения.

Рассмотрим кратко некоторые особенности данных моделей [60, 61, 147—154, 160, 161].

Модели ассоциативной памяти. Модели ассоциативной (адресно-связанной) памяти классифицируются на автоассоциативные и гетероассоциативные. В автоассоциативной памяти производится сравнение хранимого образа-эталона с поступающим в память образом по принципу (мере) сходства. В гетероассоциативной памяти строятся ассоциативные связи между парами образов. Здесь рассмотрим автоассоциативную модель памяти (модель Хопфилда) как наиболее распространенную в теории искусственных нейронных сетей.

Модель Хопфилда. В данной модели весовые коэффициенты задаются, и поэтому процесс обучения отсутствует, т. е. каждый образ хранится через связи между компонентами. При поступлении нового образа ассоциативная память вызывает хранимый образ и сравнивает с поступающим образом. Архитектура модели Хопфилда (см. рис. 4) сравнительно проста: одноуровневая модель с рекуррентными связями. Сеть имеет N модулей V_i ($i = 0, \dots, N - 1$), принимающих бинарное значение $-1/1$ (путем преобразования $(2x - 1)$ бинарного значения $0/1$). Каждому из хранимых образов P^r ($r = 1, \dots, m$) сопоставляется вектор из N бинарных компонент $(-1/1)$, соответствующих состоянию нейронной сети. Алгоритм работы нейронной сети Хопфилда имеет следующий вид:

1. Определяются веса всех образов M

$$w_{ij} = \begin{cases} \sum_{r=1}^M P_i^r P_j^r, & 0 \leq i, j \leq N - 1, \\ 0, & i = j, \end{cases}$$

где w_{ij} определяют коэффициенты связи между модулями j и i и остаются постоянными.

2. Начальное состояние $V_i(0) = I_r$.
3. Асинхронные вычисления на нейронной сети

$$V_i(t + 1) = \text{sign} \left[\sum_{j=0}^{N-1} w_{ij} V_j(t) \right] = \begin{cases} 1, & \text{если } x \geq 0, \\ -1, & \text{если } x < 0. \end{cases}$$

4. Переход к п. 3 до тех пор, пока не наступит равновесие, т. е. V_i не изменится при следующих итерациях. В состоянии равновесия вектор V_i является образом, вызванным входным сигналом I .

Хопфилдом была введена энергетическая функция (функция Ляпунова) вида $H = -\frac{1}{2} \sum_{ij} w_{ij} V_i V_j$ и сделано предположение, что хранимые образы являются

динамическими аттракторами (минимизирующие функцию энергии). Модель Хопфилда имеет два ограничения. Во-первых, ограничено количество хранимых и вызываемых образов. Оптимистическая оценка максимальной способности хранения для случайных образов составляет примерно $0,138 N$ (меньше, чем число возможных состояний 2^N). Во-вторых, связи между хранимыми образами могут привести к определению распознаваемого образа, не совпадающего с искомым. Этот эффект называется проблемой ложного локального минимума. Нейронная сеть в виде машины Больцмана позволяет преодолеть данный эффект, используя процедуру моделирования методом отжига и применения статистики Больцмана. Учет дополнительных ограничений (таких метода ортогональности образов) можно устранить указанные недостатки. Примеры применения данного подхода в теории оптимизации процессов управления можно найти в [60, 162].

Модель нейронной сети Хэмминга позволяет избежать применения квадратичных матриц весовых коэффициентов, используемых в модели Хопфилда. В этом случае построение ассоциативных связей во входном сигнале X , описывающем множество испытуемых образов, разбивается на два этапа. На первом

этапе входной сигнал сети net_i для M -мерного объекта X вычисляется по мере расстояния Хэмминга $D^H(X, P)$ между объектом X и объектом P типа

$$net_i = D^H(X, P) = \frac{1}{2} \sum_{r=1}^M P_r X_r \quad (3.8)$$

На втором этапе входной суммарный сигнал (3.8) преобразуется нелинейным отображением в out_i на основе (3.6) и передается на следующий уровень нейронов, представляющих матрицу связи с весами

$$w_{ik} = \begin{cases} 1 & \text{для } i < k, \quad i > k, \\ -e & \text{для } i = k, \quad e < \frac{M}{2}. \end{cases}$$

Данная матрица используется только для определения испытуемого образа P^i , наиболее подобного входному сигналу X . После вычисления, подобных модели Хопфилда, определяется единственный (отличный от нуля) входной сигнал net_i и максимально подобный входному сигналу X .

Модель АВМ. Двухнаправленная ассоциативная память является одноуровневой нейронной сетью и подобна сети Хопфилда. АВМ использует супервизорное обучение, и поэтому применяется испытание пар объектов (X', Y') . При этом могут быть использованы различные размерности пространств X и Y типа $m \times n$. Основная идея модели АВМ заключается в построении процедуры умножения матрицы $(m \times n)$ справа на вектор в двух направлениях: на m -мерный вектор размерности m или в транспонированном виде размерности n . На языке нейронной сети данный факт означает, что вход X может воспроизводить выход Y ; наоборот, по выходу Y можно воспроизводить вход X , используя соответствующую матрицу весов. В результате по паре сигналов (X, Y) можно конструировать другую пару (X', Y') . Основной операцией в методе АВМ является построение матрицы весов $W_{ij} = f(X_i)g(Y_j)$, где f — сигмоидальная функция вида (3.5). Для воспроизводимых K пар (X', Y') имеем

$$W_{ij} = \sum_{p=1}^K f(X'_i)g(Y'_j).$$

Итеративная процедура обучения имеет вид $(X, Y) \rightarrow W$; $XW \rightarrow Y'$; $YW' \rightarrow X''$; $(X', Y') \rightarrow W'$; $X'W' \rightarrow Y''$; $Y'W'' \rightarrow X'''$; $(X'', Y'') \rightarrow W''$ и т. д. Если на i -й итерации обнаруживается, что воспроизводимая по W' пара (X', Y') идентична паре входов: $X'W' \rightarrow Y'$, $Y'W'' \rightarrow X'$, то процесс моделирования заканчивается. В реальных приложениях формула для вычисления коэффициентов связи W_{ij} изменяется в динамическом режиме $W_{ij}^{new} = (1 - a)W_{ij}^{old} + a \sum_{p=1}^K f(X'_i)g(Y'_j)$. Сходимость данного процесса существенно зависит от параметра a (обычно равный 0,3) и m (принимается равным 0,2).

Модель обучения Кохоэна. Основная идея данной модели состоит в использовании последовательно по одному нейрону и его ближайшего пространства в топологии структуры нейронной сети. Для супервизорного обучения позиция выбираемого нейрона определяется заранее: при несупервизорном обучении выбирается нейрон с наиболее высоким уровнем выходного сигнала или нейрон j с коэффициентами весов связи с наиболее подобными компонентами для входных сигналов. После выбора нейрона матрица весов определяется в виде

$$W_{ij}^{new} = W_{ij}^{old} + a(X_i - W_{ij}^{old}),$$

где параметр a зависит от топологии расстояния до центрального нейрона и уменьшается с количеством входов. В общем случае параметр a уменьшается с увеличением топологического расстояния, но в отдельных случаях может иметь

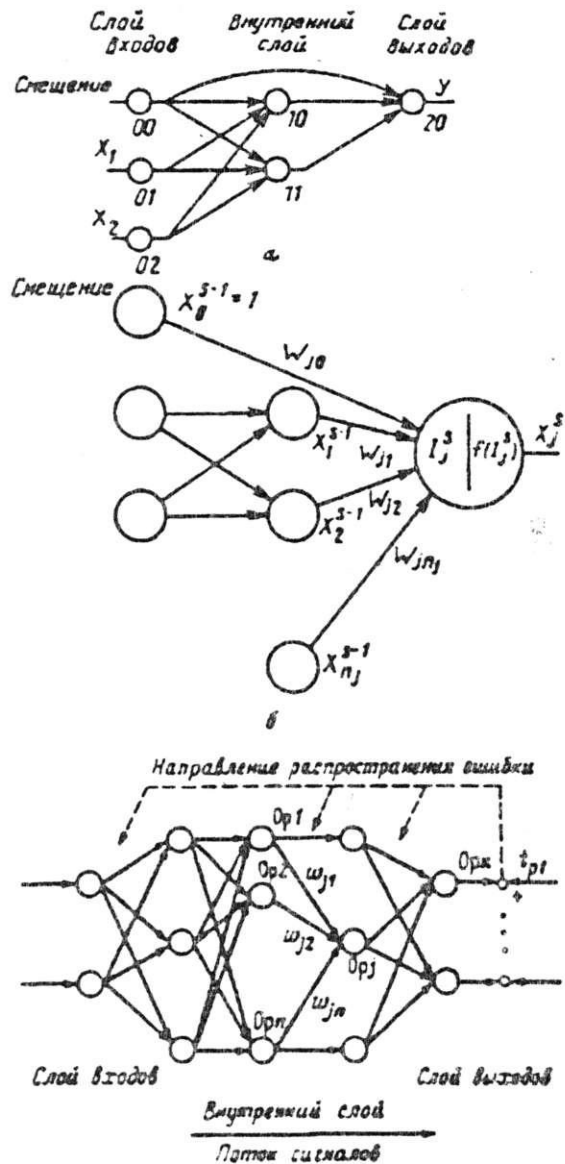


Рис. 5. Пример модели нейронной сети: а — модель Румелхарта; б — структурная схема реализации формального нейрона; в — структурная схема реализации процесса обучения методом обратного распространения ошибки

возрастающую форму [160]. Коррекция коэффициентов матрицы весов осуществляется одновременно, и далее на сети вычисляется новый входной сигнал.

Метод обратного распространения ошибки. Рассматриваемый метод разработан в [163] и является обобщением алгоритма среднеквадратичной ошибки. Нейронная сеть с обратным распространением ошибки является многоуровневой сетью с прямым распространением потока сигналов, использующей обучающий алгоритм с обратным распространением сигнала ошибки. Часто данную модель называют многоуровневым перцептроном, так как она представляет

Данный факт имеет особое значение при использовании нейронных сетей в приложениях.

Нейронные сети с радиальными функциями смещения RBFNN (Radial Bias Function Neural Networks) являются частным важным случаем многослойных нейронных сетей [189]. Соотношение «вход — выход» в виде отображения ($x \rightarrow y$) в RBFNN с m входами, выходами и N внутренними слоями (см. рис. 5, а) характеризуется в виде

$$\xi_i = g(|x - c_i|/\sigma), \quad y = C\xi, \quad i = 1, 2, \dots, N, \quad (3.10)$$

где $x \in R^m$ и является входом нейронной сети, $\xi \in R^N$ — выходные сигналы внутреннего слоя, $y \in R^r$ — выход нейронной сети; $C \in R^{r \times N}$ — матрица весовых коэффициентов; $c_i \in R^m$ и $\sigma_i > 0$ — центральное значение и ширина (или фактор разброса) модуля RBFNN соответственно. При этом часто используется евклидова или взвешенная евклидова нормы $|\cdot|$. Функция активации (сигмоидальная функция) $g: [0, \infty) \rightarrow R$ обычно выбирается в виде гауссовой функции типа $g(\beta) = \exp(-\beta^2)$ и определяет гауссовую RBFNN.

З а м е ч а н и е 2. Рассмотрим кратко обобщение приведенного алгоритма метода обратного распространения ошибки на интервально значимые данные [190]. Согласно арифметики интервально-значимых данных [191], интервал представляется нижним (левым) и верхним (правым) пределами, например $x_{\mu} = [x_{\mu}^l, x_{\mu}^u]$. Пусть $\circ \in \{+, -, \cdot, /, \div\}$ — бинарная арифметика на множестве действительных чисел. Известно [190], что если A и B — замкнутые действительные интервалы, то $A \circ B = \{a \circ b \mid a \in A, b \in B\}$ определяют бинарную арифметику на множестве всех замкнутых действительных интервалов. Операции над интервалами имеют следующий вид:

$$A + B = [a^l, a^u] + [b^l, b^u] = [a^l + b^l, a^u + b^u];$$

$$A - B = [a^l - b^u, a^u - b^l]; \quad m \cdot A = m [a^l, a^u] = \begin{cases} [ma^l, ma^u], & m \geq 0. \\ [ma^u, ma^l], & m < 0. \end{cases}$$

Рассмотрим трехуровневую нейронную сеть типа рис. 5, а, б. Как и ранее, архитектура интервально-значимой многослойной нейронной сети определяется следующим образом.

Входные модули: $O_{\mu} = X_{\mu}, \quad \mu = 1, \dots, n;$

Внутренние модули:

$$O_{\nu} = f(\text{net}_{\nu}), \quad \nu = 1, 2, \dots, p; \quad \text{net}_{\nu} = \sum_{\mu=1}^n w_{\nu\mu} O_{\mu} + \theta_{\nu};$$

Выходной модуль:

$$O_p = f(\text{net}_p); \quad \text{net}_p = \sum_{\nu=1}^p w_p O_{\nu} + \theta_p.$$

Интервально-значимые отношения «вход — выход» для сигмоидальной функции рис. 3, б принимают в рамках интервальной арифметики следующий вид.

Входные модули: $O_{\mu}^l = x_{\mu}^l; \quad O_{\mu}^u = x_{\mu}^u; \quad \mu = 1, 2, \dots, n;$

Модули внутренних слоев: $O_{\nu}^l = f(\text{net}_{\nu}^l), \quad \nu = 1, 2, \dots, p;$

$$O_{\nu}^u = f(\text{net}_{\nu}^u), \quad \nu = 1, 2, \dots, p; \quad \text{net}_{\nu}^l = \sum_{\mu=1}^n w_{\nu\mu} O_{\mu}^l +$$

$$+ \sum_{\mu=1, w_{\nu\mu} < 0} w_{\nu\mu} O_{\mu}^u + \theta_{\nu}; \quad \text{net}_{\nu}^u = \sum_{\mu=1}^n w_{\nu\mu} O_{\mu}^u + \sum_{\mu=1, w_{\nu\mu} < 0} w_{\nu\mu} O_{\mu}^l + \theta_{\nu}.$$

Выходной модуль:

$$O^* = f(\text{net}^*); \quad O^* = f(\text{net}^*);$$

$$\text{net}^* = \sum_j w_{ij} x_j + \theta_i;$$

$$\text{net}^* = \sum_j w_{ij} x_j + \theta_i;$$

Используя интервально-значимые значения O_i , сопоставляющие входному вектору-сигналу X_i , функция стоимости E определяется в виде

$$E = \sum_i \left\{ \frac{1}{2} (O_i - t_i)^2, \text{ если } |O_i - t_i| < 1, \right.$$

$$\left. \frac{1}{2} (O_i - t_i)^2, \text{ если } |O_i - t_i| > 1. \right.$$

В данном алгоритме функция стоимости определяется как максимум квадратичной ошибки между интервально-значимым выходным сигналом и целевым выходным сигналом t_i . Обучение на нейронной сети заключается в минимизации функции стоимости E . В этом случае коэффициенты весовых значений w^t и θ^t изменяются по следующим правилам: $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$ и $\Delta \theta_i = -\eta \frac{\partial E}{\partial \theta_i}$, где $\frac{\partial E}{\partial w_{ij}}$ и $\frac{\partial E}{\partial \theta_i}$ вычисляются следующим образом:

1) $\frac{\partial E}{\partial w_{ij}}$, если $|O_i - t_i| < 1$ и $w_{ij} > 0$, то

2) $\frac{\partial E}{\partial w_{ij}}$; если $|O_i - t_i| > 1$, $w_{ij} > 0$ и $t_i > 0$,

то

$$\frac{\partial E}{\partial w_{ij}} = (O_i - t_i) \cdot \frac{\partial E}{\partial O_i} \cdot \frac{\partial O_i}{\partial w_{ij}} = (O_i - t_i) \cdot \frac{\partial E}{\partial O_i} \cdot x_j.$$

Аналогично вычисляются другие варианты производных. Как ранее, вводится момент θ_i , определяющий ускорение сходимости процедуры обучения и совпадающий по форме с (3.9).

Алгоритм обучения и адаптации для нейронной сети может использовать как режим статической идентификации, так и динамический режим идентификации и управления. В последнем случае функция стоимости E определяется как функция ошибки

$$E = \frac{1}{2} \sum_i (y_i - \Phi(a_i))^2 \quad (3.11)$$

где E — оператор математического ожидания, $\Phi(a^p)$ — вектор параметров $\{a^p\}$ (см. (3.4)). Градиент от (3.11) по вектору $\Phi(a^p)$ вычисляется по формуле (3.11) для оптимизации (3.11) с помощью

$$\frac{\partial E}{\partial a^p} = -\sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial a^p} = -\sum_j \frac{\partial E}{\partial y_j} \Phi'(a^p) \cdot b_{jk} \quad (*)$$

где $\text{diag}[J_A]$ — диагональная матрица и имеет вид

$$\text{diag}[J_A] = \text{diag} \left\{ \frac{\partial y_j}{\partial a^p} \right\}, \quad \frac{\partial y_j}{\partial a^p} = \Phi'(a^p) \cdot b_{jk}, \quad j = 1, 2; \quad k = 1, 2.$$

Параметры μ_s, μ_b определяют устойчивость и скорость сходимости к оптимальным значениям. Тогда параметры прямой и обратной синаптических связей определяются как

$$a_{ji}(k+1) = a_{ji}(k) - \mu_s \frac{\partial J}{\partial a_{ji}(k)}, \quad i = 0, 1, 2;$$

$$a_{bj}(k+1) = a_{bj}(k) - \mu_b \frac{\partial J}{\partial a_{bj}(k)}, \quad j = 1, 2.$$

Параметр g , соматической функции можно определить как

$$g_s(k+1) = g_s(k) - \mu_s \nabla_{g_s} J|_{g_s(k)}.$$

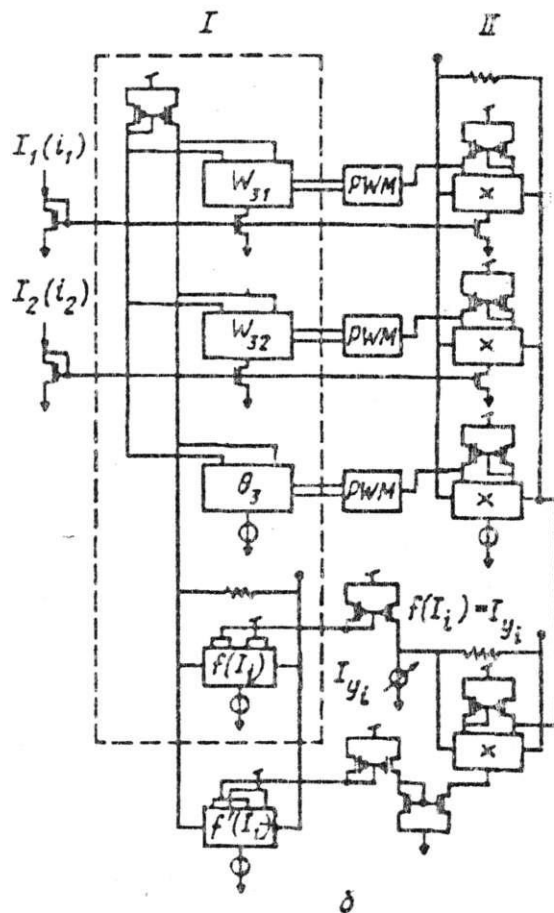
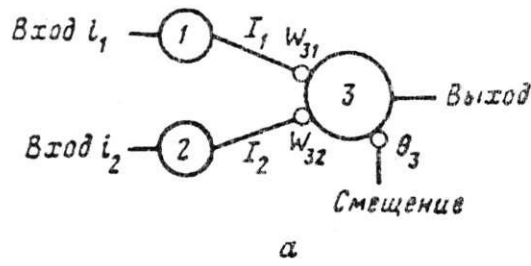
В результате получаем алгоритм обучения и адаптации моделирования нелинейных динамических систем на нейронной сети [190].

З а м е ч а н и е 3. При разработке и проектировании нейрорегуляторов для целей адаптивного управления разделяют функции контроллера и блока наблюдателя [177]. В общем случае метод обратного распространения ошибки не гарантирует устойчивость процесса адаптации, так как в параметры нейронной сети не входят характеристики динамического поведения объекта управления. Условия устойчивости адаптивного контроллера зависят от выбора начальных условий матрицы весов, параметра скорости обучения и выбора параметров сигмоидальной функции. Оптимальное соотношение между указанными параметрами устанавливается методом проб и ошибок. Для блока наблюдателя условия устойчивости, связанного непосредственно с нейронной сетью, получены в [177] для трехслойной сети.

Модели АРТ и динамические структуры нейронных сетей. Модели ART 1, 2, 3 (ART — Adaptive Resonance Theory) [192] используют идею резонансного совпадения образов и их подобия, представляют собой двухуровневую структуру прямых и обратных связей. Выбор искомого образа осуществляется методом динамического отбора и исключения не совпадающих образов. Развитие данного подхода получило в динамических структурах нейронных сетей, использующих фундаментальный принцип нейробиологии: нейроны кодируют развитие динамического поведения за счет хранения в памяти связей. В результате нейронная сеть рассматривается как многослойная структура, каждый слой имеет многоуровневую структуру и взаимодействует с другими слоями через соответствующие связи [193].

Ряд других подобных подходов рассмотрен в [147—157]. Приведенные модели могут быть объединены по единому принципу оценки близости искомого образа к эталону по расстоянию Хэмминга [161].

4. Программно-аппаратное обеспечение для НС. Разрабатываемый в последние годы программно-аппаратный инструментальный поддержки процессов моделирования на нейронных сетях получил существенное развитие и стал предметом пристального изучения. Так, в [194] было проведено сравнение пяти программных систем: Pla Net версия 5.0 (разработка University of Colorado at Boulder); Pygmalion версия 2.0 (разработка University College, Лондон); Rochester Connectionist Simulator версия 4.2 (разработка University of Rochester; Stuttgart Neural Net Simulator (SNNs) версия 1.3 и 2.0 (разработка Institute for Parallel and Distributed High-Performance Systems, University of Stuttgart, Германия). Все указанные системы реализуют метод обратного распространения ошибки. Сравнение результатов моделирования процессов распознавания образов показал, что система Pygmalion удовлетворила лучше других систем предъявляемым требованиям по точности, изменению ошибки во времени и характеристикам воспроизведения. Система Pygmalion использует операционную систему Unix и XWindow System, инсталляция на рабочей станции типа vrac-station занимает 30 мин. Связи в структуре данной системы имеют иерархический характер, нижние



уровни компонуются в кластеры, содержащие нейроны и синаптические связи. Состояния системы представляются через данные о состояниях нейронов и весовые коэффициенты синаптических связей. Помимо традиционного общего представления (слой выходов, внутренние слои и слой входов) система имеет различные гетерогенные подуровни.

Перечисленные системы успешно справляются с малым и средним уровнями сложности задач. Так, например, система SNNS может использовать до 10^4 модулей и 10^7 связей, требует около 230 Мбайт памяти для сети в 10^4 модулей и $2 \cdot 10^7$ связей. При необходимости иметь гибкие структуры для моделирования на нейронной сети рекомендуется [194] применять системы Pla Net и SNNS.

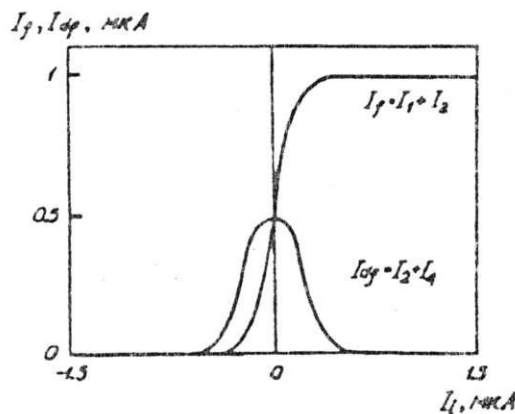
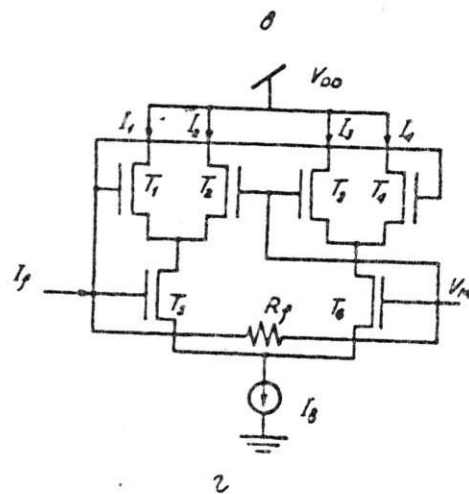


Рис. 6. Аппаратная реализация нейронной сети: а — модель двухуровневой нейронной сети; б — ее схема; в — схема дифференциального умножителя (I_4, R_f — выходные напряжение); г — вид сигмоидальной функции $f(I_i) = I_1 + I_2$ и ее производной $f'(I_i) = I_3 \cdot I_4$

Аппаратная поддержка нейронных сетей. В задачах имитационного моделирования особое значение (помимо программного обеспечения) имеет аппаратная реализация нейронных сетей для обеспечения режимов реального времени [195—203]. Анализ современного состояния данной проблемы можно найти в [196].

Здесь кратко остановимся на одном из перспективных подходов к аппаратной реализации нейронных многослойных сетей с обратным распространением ошибки в виде аналоговых чипов [202, 203].

Согласно [202, 203], наиболее перспективными для аппаратной реализации нейронных сетей являются чипы, созданные на основе оптикоэлектронных интегрированных технологий, в которых электронные связи используются для вычислений и организации памяти, а фотонные потоки в виде светового потока применяются для параллельной передачи информации. Такой подход позволяет разработать чип с оптическими связями, адаптивным механизмом для выбора синаптических связей, нейронной сетью и самоорганизацией примитивной сети на оптикоэлектронной интегральной схеме OEIC (Optoelectronic Integrated Circuit) [203].

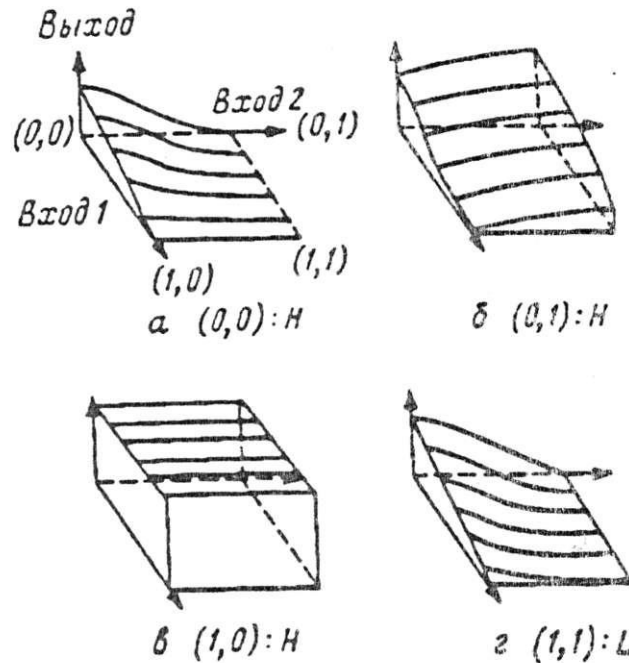


Рис. 7. Результаты моделирования на нейронной сети; а — в — сигнал обучения с уровнем «высокий»; г — сигнал обучения с уровнем «низкий»

Пример 1. Рассмотрим кратко процесс самоорганизации схемы с обратной связью для простейшей нейронной сети с адаптацией, изображенной на рис. 6. а. Алгоритм обратного распространения ошибки был применен [203] к нейронной сети (см. рис. 6, а), аппаратная реализация которой показана на рис. 6, б. Для упрощения нейронные связи были реализованы электронным способом. Сигмоидальная функция $f(I_i = \sum_j I_{ij} - \theta_i)$ и ее производная $f'(I)$ были реализованы

как I_i и I_{ij} соответственно для алгоритма обратного распространения ошибки в виде электронной схемы, изображенной на рис. 6, в на основе SiC-технологии. Соответствующая коррекция значений Δw_{ij} синаптических весов для алгоритма обратного распространения ошибки в электронной схеме с умножителем (см. блок П рис. 6, д): $\Delta w_{ij} = a(I_{ij} - f(I_i)) f'(I_i)$, где I_{ij} — сигнал обучения, a — постоянная.

Результаты эксперимента [203] показаны на рис. 7. Начальным требуемым значением выхода была величина «низкий» (L — Low) для всех уровней входов 1 и 2. Далее сигналы обучения последовательно изменять для проверки работоспособности схем следующим образом. Выходы обозначались сигналами обучения в виде «высокий» (H — high) для входов 1 и 2 как (0,0) (см. рис. 7, а); (0,1) (см. рис. 7, б) и (1,0) (см. рис. 7, в). Для сигналов 1 и 2 в сочетании (1,1) принималось требуемое значение «низкий». Экспериментальные значения выходных сигналов при заданных сигналах обучения показаны на рис. 7. Представленная схема обучения параллельно обрабатывает входные сигналы и независимо от изменения одного из сигналов приводит к требуемому значению L в виде (1,1).

В более общем случае вычисления на нейронной сети сводятся к операции векторно-матричного умножения для которых разработаны по GaAs/Al Ga As-технологии два типа оптикоэлектронных устройств; оптический обучающийся нейрон и искусственная сетчатка [202]. В основе данных устройств лежит фотодетектор с переменной чувствительностью WSPD (Variable Sensitivity Photodetector по MSN-технологии (металл — полупроводник — металл). Общая площадь фоточувствительности в VSPD составляет $100 \times 100 \text{ мкм}^2$. Были раз-

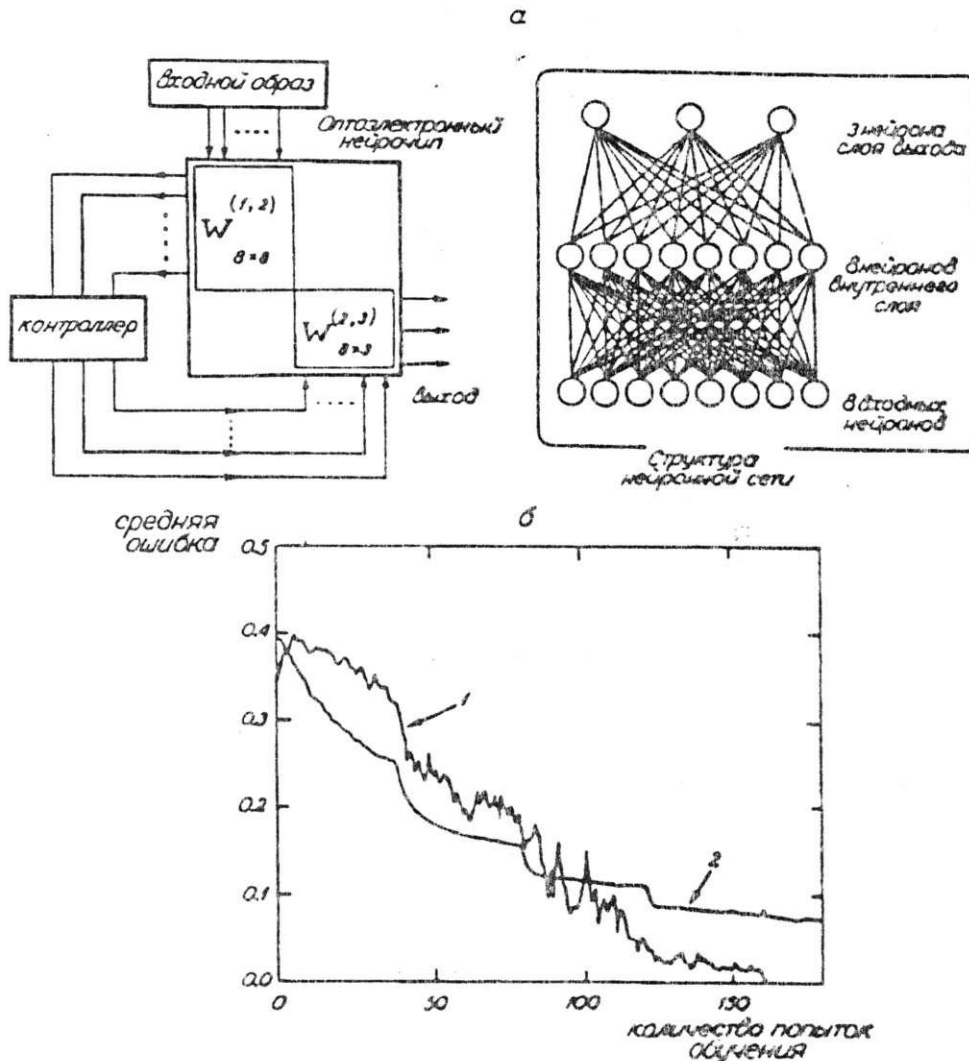


Рис. 8. Структура многослойной нейронной сети (а) и результаты моделирования средней ошибки при классификации 14 образов (б): 1 — натуральный эксперимент; 2 — имитационный эксперимент

работаны два типа оптических вектор-матричных умножителей на двумерных VSPD-матрицах. В частности, нейрон имеет внутреннюю аналоговую память для синаптических весов и объединяет 32 нейрона. Общий размер нейрончика составляет площадь $6,3 \times 6,3 \text{ мкм}^2$. Экспериментальные результаты для нейрончика с 32 нейронами (32×32 синаптических чипа) показали, что каждый VSPD-элемент имеет в двумерной матрице индивидуальный доступ, и хранение сигналов осуществляется на основе матричной адресной схемы.

Пример 2. Экспериментальная проверка работоспособности оптоэлектронного нейрончика проверялась на процессах обучения трехуровневой модели с алгоритмом обратного распространения ошибки, изображенной на рис. 8. Для корректной аппаратной реализации трехуровневой сети чип пространственно разделен на две области: область «уровень входов — уровень внутреннего слоя» (1,2) и область «уровень внутреннего слоя — уровень выходов» (2,3). Здесь 1, 2, 3 означают уровни входов, внутреннего слоя и выходов соответственно. Число нейронов в этих слоях принято $M^I = 8$, $M^II = 8$, $M^III = 3$ соответственно. Общее

количество светонизлучающих диодов и VSPD-элементов в данном оптоэлектронном чипе составляет 16 и 88 соответственно. Во время процесса обучения синаптические веса хранятся в ячейках VSPD. Нелинейные процессы обрабатываются параллельно в II (слой $N^{(2)} + N^{(3)}$) внешних электронных схемах. В процессе моделирования вычислялась средняя ошибка в зависимости от количества попыток обучения. Использовался оптоэлектронный нейрочип в задаче классификации 14 образов, предъявляемых к испытанию с 8 — разрядным кодированием для разделения на 3 категории. В фазе обучения при некорректном выходном сигнале синаптические веса изменялись согласно алгоритму обратного распространения ошибки. В результате величины напряжений, соответствующие сигналам ошибки, передавались в VSPD-элементы для изменения хранимых синаптических весов. Результаты эксперимента приведены на рис. 8, б. Скорость обучения ρ и демпфирующий коэффициент α принимались 0,4 и 0,7 соответственно. Входные образы корректно классифицировались по трем категориям уже после 100 попыток обучения. Дальнейшее развитие оптоэлектронных чипов предполагает плотность интеграции более чем 2000 нейронов/см² и скорость обработки более 1 TCPS. Реализация искусственной сетчатки с высокой разрешающей способностью на (1000 × 1000) VSPD — элементах ожидается в ближайшее время [202]. В настоящее время сетчатка с 64 × 64 VSPD-элементами успешно справляется с распознаванием графических образов различной степени сложности [202].

Рассмотренные в данном разделе модели нейронных сетей обладают рядом особенностей. Одна из них состоит в том, что данные модели ориентированы на определенную предметную область (например, задачу поиска оптимального решения удобнее реализовать на модели Хопфлда, а задачи классификации, анализа и интерполяции — на модели Румелхарта (см. рис. 4). Наличие динамического или статического характера нейронных сетей накладывает дополнительные ограничения на время работы вычислительных средств. В зависимости от требования наличия или отсутствия режима реального времени выбираются различные аппаратные средства поддержки реализации нейронной сети. Приведенные примеры реализации многослойных нейронных сетей в виде аналоговых оптоэлектронных микросхем свидетельствует о широких возможностях использования подобных устройств непосредственно в контуре управления для реализации таких режимов, как управление с обучением в нечетких самоорганизующихся регуляторах при формировании соответствующей базы знаний.

В соответствии с принципами проектирования систем управления с различным уровнем интеллектуальности [39] выбор уровня интеллектуальности должен соответствовать уровню применяемых исполнительных устройств. Использование нейронных сетей с соответствующей аппаратной поддержкой позволяет методами имитационного моделирования установить требуемое соответствие.

5. Генетические алгоритмы в имитационном моделировании. Традиционные подходы к решению задач имитационного моделирования основывались на вычислительных методах, хорошо приспособленных для поиска вариантов нахождения градиентов (а в ряде случаев и вторых производных) вычисляемых функций, заданных численно или аналитически. Пространство поиска предполагалось унимодальным, размерность пространства поиска считалась малой. Разработанные алгоритмы обладали робастностью для достаточно широкого класса оптимизируемых функций, широко используемых в прикладных задачах. В последние годы были предложены различные методы, позволяющие сократить вычислительные трудности при поиске оптимумов функций, являющихся невыпуклыми или разрывными. Один из таких подходов основан на методе случайного поиска. Дальнейшее развитие этот подход получил в проблемно-независимом генетическом алгоритме (ГА). Данный ГА является похвальной процедурой, использующей аналогию с принципами генетики. В дальнейшем мы воспользуемся некоторыми основными положениями теории генетических алгоритмов на качественном уровне изложения. Более подробные сведения можно найти в [204—211].

Как уже отмечалось, ГА является примером еще одной (кроме моделей нейронных сетей) разновидностью биологической аналогии, использующей идею дарвиновской эволюции на основе принципа генерации, тестирования и отбора наиболее вероятных популяций. В этом случае воспроизводится хромосома в виде цепочки из l позиций; каждая позиция — ген, который принимает значения (allele) в соответствии с алфавитом $A = \{0, 1, * \}$. Выбор основан на ГА в виде порождения правил $[H, V]$, где H — мощность алфавита, V — используемая в структуре хромосом информация. Поясним вводимые понятия предварительным примером.

Пример 3. Допустим, что задан класс обучающих систем автономного робота на множестве объединенных событийных задач. Опознание события осуществляется датчиками измерения и мониторингом, а процесс обучения — через обратную связь по оценке качества опознания события. В результате имеем модель принудительного обучения с закреплением оценки качества событий (так называемая reinforcement learning system) [212]. Предположим, что в процессе обучения измерения осуществлены с помощью алфавита A в виде $*1*$: $001 \rightarrow 010$. В этом случае имеем два условия $*1*$: 001 . Второе условие порождено измерением 001 . Первое же условие порождено некоторым измерением с 1 во второй позиции. Символ $*$ означает оператор «безразлично». Тогда оба символа $(0,1)$ могут занимать позицию и вторая генерация хромосомы 010 пересылается в список измерений события.

Следует отметить следующие особенности ГА, принципиально отличающие его от традиционных методов поиска оптимума: 1) в работе ГА используется кодирование множества параметров, а не сами параметры; 2) поиск осуществляется на основе популяции точек, а не в точках пространства поиска; 3) в ГА применяются вероятностные, а не детерминированные правила поиска.

Таким образом, в ГА кодируются исходные параметры в цепочки конечной длины, итерациями воспроизводится и тестируется популяция цепочек. Такой подход аналогичен воспроизведению естественных популяций в биологических системах. Этот подход принципиально отличается от традиционных методов анализа перехода от одной точки пространства поиска к другой на основе заданных правил. Следовательно, ГА является инструментарием для поиска решения на классе задач, в котором используется структура знаний предыдущих решений с приоритетом для кандидатов возможных решений.

Пример 4. Допустим, что в задаче оптимизации осуществляется отображение из множества $K_{\text{низ}}$ возможных параметров K_1 с малой вероятностью во множество $K_{\text{выс}}$ возможных параметров K_2 с высокой вероятностью событий. Данное отображение использует 10 — разрядное кодирование для параметров K_1 и K_2 . В этом случае кодирование цепочки 0000000000 соответствует отображению в $K_{\text{низ}}$ и 1111111111 — отображению в $K_{\text{выс}}$. Между $K_{\text{низ}}$ и $K_{\text{выс}}$ существует линейное отображение. Оба 10 — разрядные представления объединяются в 20 — разрядное представление цепочки (например, для проектирования регулятора с коэффициентами усиления в обратной связи K_1 и K_2). Такая 20 — разрядная цепочка представляет $2^{20} = 1048576$ альтернативных решений. Поэтому в ГА отбор осуществляется по критерию оптимума так называемой функции пригодности. Популяция эволюционирует во времени, конкурируя с другими воспроизводимыми популяциями (с оценкой по функции пригодности) и контролируются отклонения (рекомбинацией и мутацией) функции пригодности. Теоретический анализ показывает [204], что генетический алгоритм использует структурированное знание при исследовании возможных новых решений на всем пространстве решений и осуществляет концентрацию решений в наиболее вероятных подобластях всего пространства решений. Рассмотрим данную особенность ГА более подробно. Основная идея ГА проиллюстрирована на рис. 9. В моделях генетического алгоритма воспроизводится структурированное знание о возможных решениях рассматриваемой проблемы. Начальная популяция может быть воспроизведена с использованием знания о возможных решениях. В общем случае

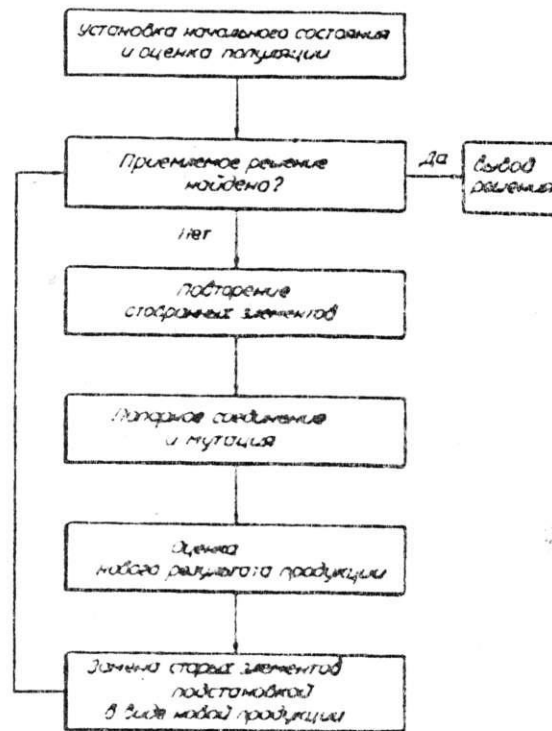


Рис. 9. Блочная схема генетического алгоритма

(в ситуации, когда знания о возможных решениях отсутствуют) популяция является случайной выборкой из исследуемой области решений. Каждый член популяции оценивается и представляется через относительную меру пригодности в качестве возможного решения. После того как каждая структура в популяции оценена, воспроизводится новая популяция в два этапа. На первом этапе структуры в текущей популяции отбираются для репликации (копирования) на основе их относительных мер пригодности (соответствия). Наиболее пригодные структуры отбираются каждый раз для репликации, а менее пригодные структуры исключаются из рассмотрения. В отсутствие других механизмов результирующий принудительный отбор приводит к все более и более пригодным структурам популяций, содержащих больше знаний о возможно пригодных решениях. На втором этапе отобранные структуры видоизменяются с помощью так называемых генетических операторов к форме нового множества структур для осуществления процесса эволюции. Первый из двух операторов носит название оператора пересечения (crossover operator), который осуществляет попарное объединение исходных (родительских) структур для формирования двух новых подобных структур (потомков — offspring). Существует достаточно большое количество форм операторов пересечения. Например, если исходные популяции имели вид (a_1, a_2, a_3, a_4) и (b_1, b_2, b_3, b_4) , то в результате применения оператора пересечения в виде замены сегментов цепочки получим (a_1, a_2, b_3, b_4) и (b_1, b_2, a_3, a_4) . В воспроизводимых новых структурах для испытаний на тест пригодности оператор пересечения осуществляет жеребьевку на основе информации, представленной в структурах с текущей базой знаний. В процессе применения итераций отбора может происходить потеря специфической информации. В этом случае оператор пересечения не может воспроизводить новых структур, в которых эта информация могла бы содержаться. Поэтому применяется второй из упомянутых операторов — оператор мутации (mutation operator). С его помощью осуществляется коррекция одной или более

компонент выбранной структуры, обеспечивая путь для введения новой информации в базу знаний. Как и для оператора пересечения, существует достаточно широкий выбор вида оператора мутации [205]. В большинстве случаев мутация служит как вторичный поисковый оператор, гарантирующий возможность охвата всех точек в пространстве поиска решений. Результирующие популяции — потомки оцениваются и возвращаются в исходную популяцию, замещая другие элементы. Специфические решения относительно выбора количества восстанавливаемых членов во время каждой итерации и отбираемых членов для замещения определяют область выполняемых альтернатив.

Любой простейший ГА проявляет информационные возможности обработки знаний.

Рассмотрим, как ГА извлекает и использует информацию, содержащуюся в популяции. Предположим, что в момент t была воспроизведена популяция (011010) с оценкой пригодности (ОП) = 1, (100111) с ОП = 0; (101001) с ОП = 2; (110010) с ОП = 0; (011000) с ОП = 1; (000110) с ОП = 3; (000111) с ОП = 1; (110110) с ОП = 0. В ГА применяется правило отбора типа «количество продукции — потомков в данной структуре пропорционально структурной относительной пригодности». Используя данное правило ГА осуществляет следующую генерацию структуры (до применения пересечения и мутации): (011010) с ОП = 1; (101001) с ОП = 2; (101001) с ОП = 2; (101001) с ОП = 2; (011000) с ОП = 1; (000110) с ОП = 3; (000110) с ОП = 3; (000110) с ОП = 3; (000111) с ОП = 1. Рассмотрим частный образ (00*****), представляющих все цепочки, начинающиеся с 0. (Как и ранее, оператор * означает оператор «безразлично».) Популяция в момент t имела 4 такие цепочки, называемые схемой. Оценка пригодности данной схемы составляет $(1 + 1 + 3 + 1)/4 = 1.75$, что соответствует 50% от средней ОП = 1. Но и количество таких цепочек в новой схеме также на 50% больше, чем в предыдущей популяции. Этот факт устанавливает так называемая теорема о схемах ГА [204]: развитие экземпляров от данной схемы примерно пропорционально установленной относительной ОП данной схемы.

Нетрудно заметить, что такое же соотношение выполняется между устанавливаемой относительной ОП и скоростью развития порождения экземпляров. Генетические операторы (пересечения и мутации) могут изменить это соотношение для многочисленных образов. Но основной эффект о связи между количеством образов (экземпляров) и более высокой относительной ОП сохраняется в популяции (в то время как экземпляры с более низкой относительной ОП не воспроизводятся). Способность к эффективному поиску на большом пространстве экземпляров без непосредственных вычислений и сохранения статистики, связанной с экземплярами, называется в теории генетических алгоритмов неявным параллелизмом ГА. Особенность ГА состоит в том, что осуществляется тестирование не отдельных структур, а в способности использовать самым высоким эффективным способом информацию о пригодности очень большого количества структурных экземпляров. Это дает возможность осуществлять целенаправленно исследование и поиск в пространстве решений, в котором основное внимание уделяется областям, содержащим структуры с постоянно высокой пригодностью. База знаний широко распространена на пространстве, позволяя предотвратить поиск в локальном оптимуме.

В основу теории ГА [204] была положена главная идея о генетической природе процесса поиска. Для реализации поисковых процедур вначале использовалось двоякое представление, которое не является единственно возможным. Более того, анализ показал, что [213] существует целый ряд конкретных разработок экспериментальных систем, в которых используется другой подход. Так, система Genosop и OOGA (Object-Oriented Genetic Algorithm — объективно-ориентированный ГА) представляют хромосомы в виде векторов. Система Samuel [213] содержит обучение стратегиям поиска для многозадачных проблем. В этом случае стратегии обучения выражены в виде множества правил «если..., то...». Хромосомы в данном ГА выражаются в виде (правило А, правило Б, правило В,

правило G, ..., правило H). В [208] разработан другой метод, названный генетическим программированием, в котором используются выражения языка Лисп. При использовании не двоичного представления приходится часто переопределять генетические операторы. Так, например, в представлении на языке Лисп пересечение содержит видоизмененные правильно определенные выражения между двумя структурированными графами. В системе Samuel оператор пересечения содержит модификацию обмена эвристически выбранных правил между двумя стратегиями.

Основные операторы ГА. Простейший ГА имеет три оператора: 1) репродукция; 2) пересечение; 3) мутация. ГА начинается по закону случайных чисел воспроизводить популяцию из N цепочек, каждая длиной m . Если представить, что каждая цепочка представляет закодированным образом одно из возможных решений рассматриваемой проблемы (в частности, поиск всех нечетких функций принадлежности для базы знаний нечеткого контроллера), то каждая цепочка кодируется в соответствии с характерными признаками искомым параметров (нечетких функций принадлежности). Параметры пересылаются в математическую модель исследуемой системы, и оценивается значение пригодности. Данная ОП может быть основана на качестве решения, заложенного в данную цепочку (или, например, может быть представлена в виде меры использования нечетким контроллером ассоциативно воспроизведенных функций принадлежности в управлении системой). ОП затем используются в новой популяции. Эта новая генерация должна содержать более эффективные, например, функции принадлежности. Новые цепочки кодируются, оцениваются и трансформируются, используя три генетических оператора. Процесс продолжается до тех пор, пока не будет достигнута сходимость или не будет найдено приемлемое решение. Репродукция и пересечение придают ГА статус логического «большинства» в его возможности поисковых функций. Третий оператор, мутация, усиливает возможность ГА найти субоптимальные решения. Образно говоря, мутация выполняет жизненную функцию в ГА и имеет весьма малую вероятность события (что составляет примерно одну мутацию на тысячу позиций цепочки) и является вторичной по отношению к репродукции и пересечению. В целом ГА использует принцип отбора Дарвина со случайным изменением информации и знания, содержащихся в предыдущих решениях для увеличения эффекта механизма поиска по мощности и скорости.

З а м е ч а н и е 4. Проанализируем некоторые дополнительные особенности работы генетических операторов выбора, пересечения и мутации. Как отмечалось, выбор популяции является процессом, когда старая цепочка переводится преобразованием в новую цепочку в зависимости от значения относительной пригодности. В результате при высокой оценке степени пригодности воспроизводится больше экземпляров при следующей генерации. При этом выбираемые экземпляры могут быть отобраны по разным критериям. Одним из простых способов такого выбора является отбор цепочек с высокой степенью пригодности F пропорционально высокой вероятности P выбора (отбора на основе принципа рулетки [205]) в виде $P(\text{выбора}) = F / \sum F$, i — индекс цепочки. Стратегия, в которой цепочки с высокой степенью относительной пригодности воспроизводят большое число экземпляров, отражает формальную сторону концепции ГА. Существует совместно с приведенной стратегией методы, позволяющие изменять (чистый выбор) или сохранять в следующей эволюции хорошие цепочки.

Простейший оператор пересечения продолжает выбор в три этапа. На первом этапе осуществляется попарное объединение цепочек по случайному закону. На втором этапе номер позиции i по каждой паре цепочек выбирается также по случайному закону. И, наконец, на третьем этапе с заданной вероятностью пересечения попарные цепочки подвергаются операции пересечения над позицией i вдоль цепочки. Допустим, что две цепочки закодированы в виде

$$X = X_1 X_2 X_3 X_4 X_5 (00011)$$

$$Y = Y_1 Y_2 Y_3 Y_4 Y_5 (11100)$$

Если, например, жребий выбрал позицию $l = 3$, то в результате применения оператора пересечения образуются две новые цепочки X' и Y' вида

$$X' = Y_1 Y_2 Y_3 X_4 X_5 (11111)$$

$$Y' = X_1 X_2 X_3 Y_4 Y_5 (00000).$$

Процесс пересечения также является случайным событием, и совместно с процессом выбора создает основу для эффективного обмена информацией и выбора высокого качества решений. Поэтому вид операторов выбора и пересечения определяют выразительную мощность поиска в ГА.

Третий оператор, мутации, изменяет позиции случайным образом в цепочке (на основе вероятности мутации). В бинарном коде осуществляется замена 1 на 0 и наоборот. Основным назначением введения оператора мутации является избежание возможности попадания в локальные экстремумы при поиске глобального экстремума. В сочетании с двумя первыми генетическими операторами воспроизводится по аналогии с генетикой работа ГА.

Следует отметить, что исполнение ГА обычно оценивается в режимах «on-line» и «off-line». При режиме «on-line» оценка ГА осуществляется по мере пригодности, усредненной по всему множеству испытуемых в поиске цепочек. Режим «off-line» использует меру усреднения только на множестве цепочек с высоким уровнем относительной пригодности. Исследования [214, 215] показали, что популяции воспроизводятся размером от 30 до 100; параметр пересечения от 0.6 до 0.45 и мутации от 0.001 до 0.01.

Наличие эффекта робастности в технике поиска при работе ГА объясняется приведенной теоремой о схемах и создает основу для вычислений границы роста полезных копий (которые часто называют строительными блоками — building blocks). Фундаментальный принцип ГА использования таких экземпляров с максимальной высокой полезностью может быть нарушен в силу работы кодирующего механизма. Данная проблема известна как теорема «обманывающего» ГА (GA — deceptive problems) [216]. Оказалось, что данная проблема может быть решена введением нового класса стохастических ГА под названием беспорядочного (messy) ГА [216, 217].

Механизм оценки ГА. Для понимания работы ГА необходимо в первую очередь иметь механизм оценки роста различных схем, содержащихся в популяции. Популяция размером N содержит схем в количестве от 2^l до $N \cdot 2^l$, где l — длина цепочки. Порядок схемы $O(H)$ определяется количеством 0 и 1 в схеме, а определяющая длина $\delta(H)$ схемы вычисляется как расстояние между первым и последним геном в хромосоме. Так, например, для схемы $A1 = 1***1$ имеем $O(A1) = 2$ и $\delta(A1) = 4$.

Согласно [204, 205] нижний уровень короткой по длине схемы, обладающей степенью пригодности (building blocks) возрастает по экспоненте в популяции. Количество схем, воспроизводимых в популяции из l цепочек, определяется при этом как $O(l^l)$. Такая оценка вычислительной мощности дает ГА существенные преимущества перед традиционными методами [116].

Пример 5. Рассмотрим структурную схему системы управления, представленной на рис. 10, а, и пространство поиска, ассоциированного с задачей параметрической оптимизации (рис. 10, б). Согласно рассмотренному ранее примеру, проектирование контроллера в области параметров (K_1, K_2) сводится с точки зрения применения ГА к 20 — разрядному представлению цепочки. Для решения данной задачи был выбран простейший ГА со следующими операторами: 1) выбор по принципу рулетки; 2) простое пересечение; 3) мутация. В этом случае каждая 20 — разрядная цепочка представляет одну из возможных реализаций обратной связи. На рис. 11 представлены результаты моделирования поиска решения с помощью представленного алгоритма со следующими параметрами ГА: размер популяции — 30; вероятность пересечения — 0.95 и мутации — 0.01. Из результатов моделирования видно, что пространство поиска мультимно-

а

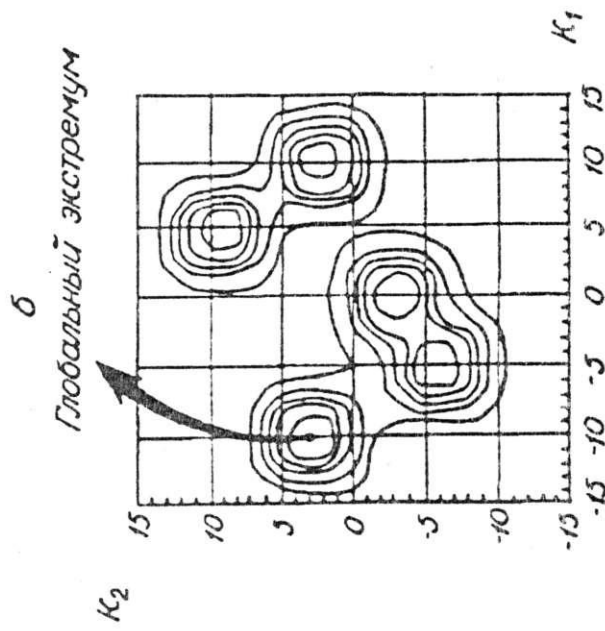
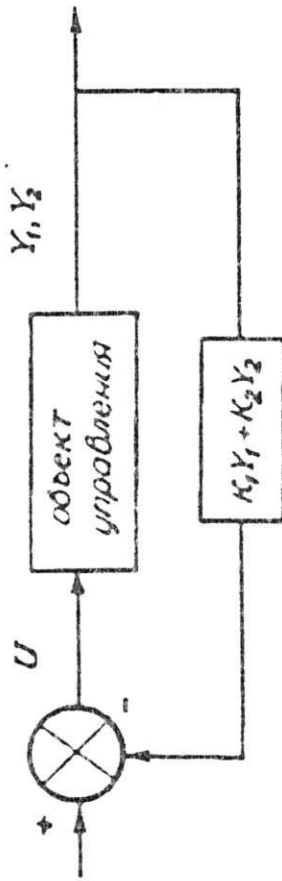


Рис. 10. Оптимизация цепи обратной связи в системе управления: а — структурная схема процесса управления; б — поиск глобального экстремума коэффициентов усилителя (K_1 , K_2) цепи обратной связи

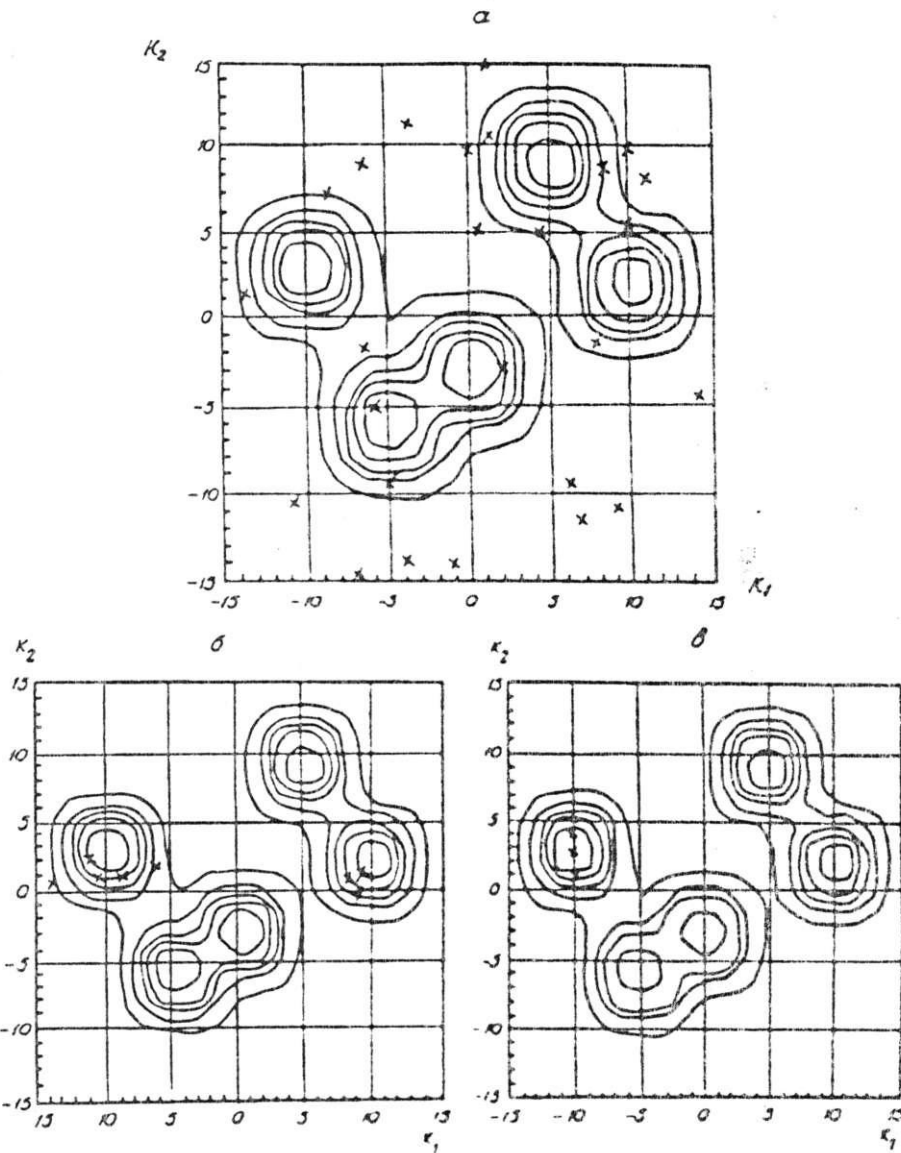


Рис. 11. Результаты моделирования процесса поиска глобального экстремума (K_1 , K_2) с применением генетического алгоритма: а — 1 итерация; б — 10 итераций; в — 30 итераций

дальное; поиск в ГА основан на вероятностном распределении точек поиска, при увеличении уровня отбора сами точки поиска группируются у локальных оптимумов, а при дальнейшем увеличении уровня отбора поиск приводит к искомому глобальному оптимуму.

6. Особенности применения ГА. Многочисленные исследования в области построения структур ГА посвящены, по существу, следующим вопросам [217]: 1) разработке строительных блоков (СБ) как основы ГА; 2) выбор адекватных начальных СБ, 3) определение условий гарантии роста СБ; 4) выделение задач, не трудных для СБ; 5) обеспечение достоверных решений на СБ; 6) определение условий, для которых решения на СБ получаются наилучшими.

Одним из важных условий успешного применения ГА является также обеспечение процесса поддержки различия распределения значений генов и хромосоме

при поиске в пространстве решений. Это объясняется необходимостью избежания эффекта «преждевременной» сходимости к локальному оптимуму. Поэтому во избежание таких явлений стремятся оценить различие (разнообразие), воспроизвести разнообразие и предотвратить потерю разнообразия. Исследование такой проблемы сводится к задаче оптимизации численных переменных X_1, \dots, X_N в терминах эволюционной функции пригодности $F(X_1, X_2, \dots, X_N)$. В отличие от традиционного подхода, когда бинарное кодирование представляло цепочку в виде хромосомы со значениями гена (allele) в виде 1 или 0 в [218] хромосома рассматривалась как массив $X[i]$ ($i = 1, \dots, N$), а значения гена (allele) принимались из интервала $[-10, 10]$. Популяция хромосом представляется двумерным массивом $X[q][i]$, где i индексирует месторасположение переменной, а q индексирует индивидуумы. Постоянный оператор пересечения используется для согласования двух индивидуумов. $N/2$ цепочек по случайному закону выбирается для попарного сравнения генов. Мера разнообразия определяется как оценка «потери числа генов» для бинарного представления цепочек [205]. Для введенного представления цепочек рассмотрим понятие гистограммы $\text{Hist}[n][i]$, определяющее количество генов в цепочке i из данной популяции с разбивкой интервала $[-10, 10]$ на 7 подынтервалов $[-10, -8], [-7, -5], \dots, [-1, 1], \dots, [8, 10]$ с индексом n . Тогда

$$\text{Hist}[n][i] = \sum_{q=1}^{Q_{\max}} s(q, n, i), \quad (6.1)$$

где

$$s(n, q, i) = \begin{cases} 1, & \text{если } 3n - 13 \leq X[q][i] \leq 3n - 11, \\ 0, & \text{в остальных случаях.} \end{cases}$$

Величина Q_{\max} используется для оценки размера популяции (полного числа хромосом). Тогда оценка потери генов $\text{AL}[i]$ в цепочке i определяется как

$$\text{AL}[i] = \sum_{n=1}^7 (\text{количество нулей в Hist}[n][i]). \quad (6.2)$$

Полное количество потери генов TAL определяется как

$$\text{TAL} = \sum_{i=1}^N \text{AL}[i]. \quad (6.3)$$

В соответствии к мере Hist , мера исключительности Rareness индивидуума определяется как

$$\text{Rareness}[q] = \sum_{i=1}^N \text{Hist}[n(q, i)][i], \quad (6.4)$$

где $n(q, i) = \{13n - 13 \leq X[q][i] \leq 3n - 11\}$. Выражение (6.4) оценивает количество индивидуумов в Hist . При этом мера Rareness мала, когда индивидуум исключительно редок. Гены всех хромосом в популяции индивидуумов попадают в области оценок исключительности. Эта мера может использоваться для оценки потери разнообразия. Для введения этой меры необходимо исследовать оператор мутации. В бинарном представлении оператор мутации используется для инверсии (0,1). В таком представлении мутация обеспечивала требуемую сходимость с вероятностью 0.01. Для обеспечения полной сходимости необходимо изменить определение оператора мутации [218]. Для этого случайная целочисленная величина d выбирается из интервала $[-Dw, Dw]$ и добавляется к ранее определенной величине $X[q][i]$. Данная операция проводится со всеми хромосомами ($i = 1, \dots, N$), когда индивидуумы подвергаются действию оператора мутации. Размах оператора мутации Dw используется для отладки эффекта рандомизации.

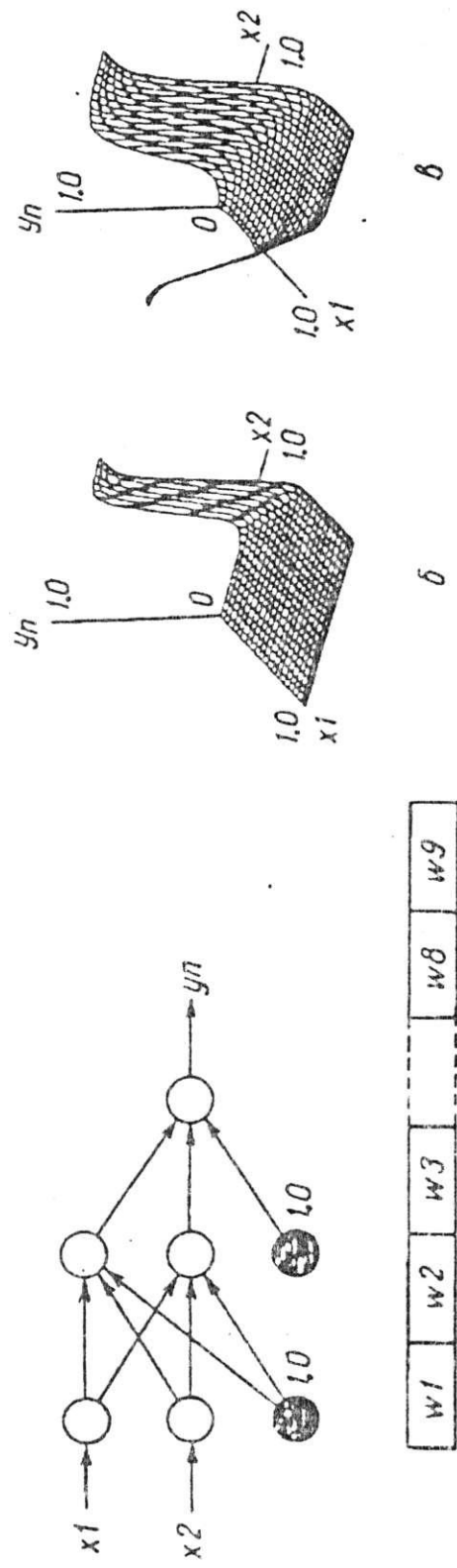


Рис. 12. Обучение на нейронной сети схемы типа «Исключение ИЛИ» с использованием генетического алгоритма. а — схема обучения; б — локальный оптимум; в — глобальный оптимум

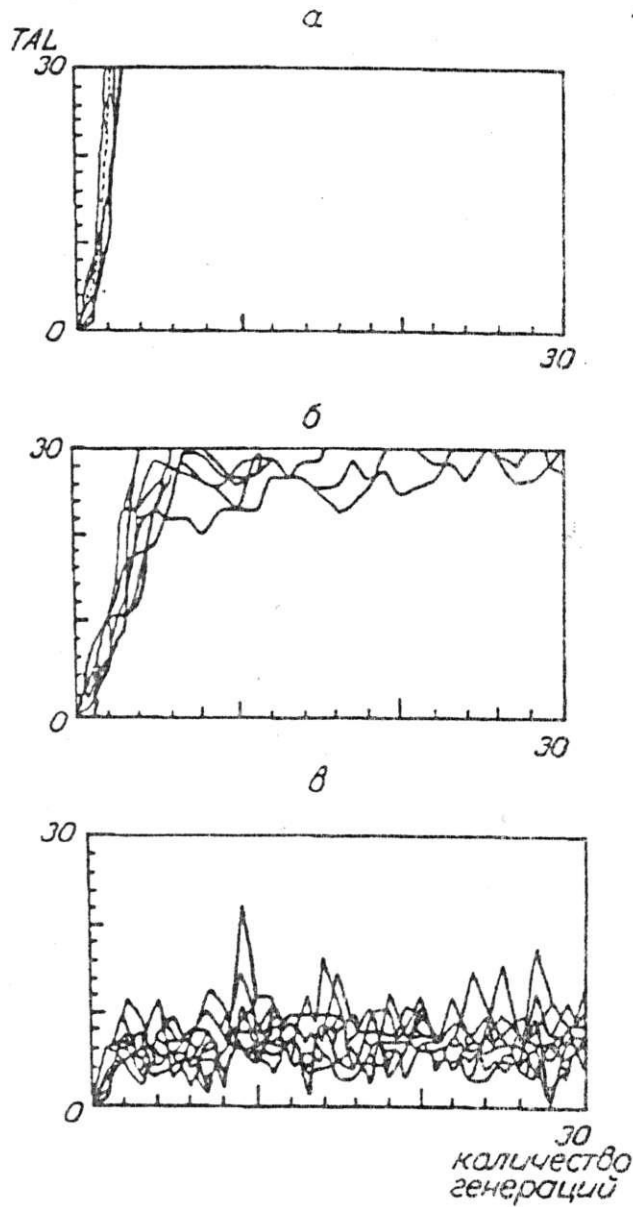


Рис. 13. Моделирование процесса поддержки эффекта разнообразия: а — мутация и масштабирование коэффициента соответствия отсутствуют, степень выхода из локального оптимума 0%; б — влияние только мутации ($D_w = 3$), степень выхода из локального оптимума 50%; в — влияние мутации ($D_w = 3AL[i]$) и масштабирования коэффициента соответствия, степень выхода из локального оптимума 80%

При большом D_w эффект эквивалентен бинарному представлению, при конечном D_w появляется возможность регулировать эффективное воспроизведение исключительности генов. Соотношение между требуемой сходимостью и сохранением разнообразия как раз и регулирует мера размаха оператора мутации в каждой хромосоме в виде

$$D_w = 3AL[i]. \quad (6.5)$$

Выражение (6.5) объединяет меры размаха оператора мутации и потери количества

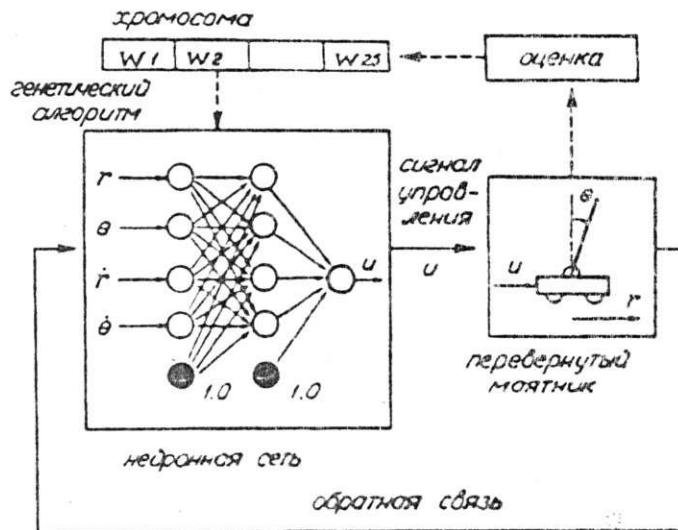


Рис. 14. Структурная схема управления нелинейным объектом

генов так, что мутация эффекта при обнаружении потери количества генов, воспрепятствуя потере исключительных индивидуумов. В добавление к мере размах оператора мутации изменение меры пригодности определяется как

$$\{\text{Новая мера пригодности } q\} = \{\text{пригодность } q / \text{Rareness } [g]\}. \quad (6.6)$$

Проиллюстрируем особенности данного подхода двумя важными примерами.

Пример 6. Рассмотрим структуру нейронной сети на рис. 12, а, используемой при моделировании схемы «Исключительное ИЛИ» (XOR). В этом случае сеть имеет три уровня и объединяет семь нейронных модулей. Сигмоидальная функция

$$0 = \frac{1 - \exp\{-i\}}{1 + \exp\{-i\}}. \quad (6.7)$$

Для данной сети имеем девять весовых коэффициентов связи, которые кодируются и представляются в виде хромосомы (см. рис. 12, а). Линейные связи между уровнями представляют описание модели в форме локальных оптимумов (рис. 12, б), а глобальный оптимум достигается при выходе хромосомы из локального оптимума (т. е. учитывается нелинейная природа связей в нейронной сети). Эффективное применение ГА оценивается для каждой хромосомы величиной V в виде

$$V = \{1y_n - \text{XOR}(X1, X2)\}^{-1}. \quad (6.8)$$

где суммирование проводится по четырем случаям входных сигналов (1,1), (1,0), (0,0), (0,1). На первом этапе ОП вычисляется непосредственно по мере V из (6.8), а затем применяется выражение (6.6). Результаты моделирования для 10 независимых прогонок представлены на рис. 13, иллюстрирующие эффект операции мутации и учета изменения меры пригодности в мере TAL (полного числа потерь генов). В случае, когда не учитываются мера разнообразия, TAL возрастает быстро (в предположении, что все хромосомы находятся в локальном оптимуме) и ни одна из 10 прогонок не ограничивает ее рост. Когда применяется мутация, TAL достигает своего ограниченного максимума (см. рис. 13, б), а при добавлении меры изменения пригодности имеет место эффект стабилизации.

Пример 7. Рассмотрим теперь задачу управления нелинейным неустойчивым объектом, на котором проиллюстрируем эффективность методологии проектирования интеллектуальных систем управления на основе применения ГА и

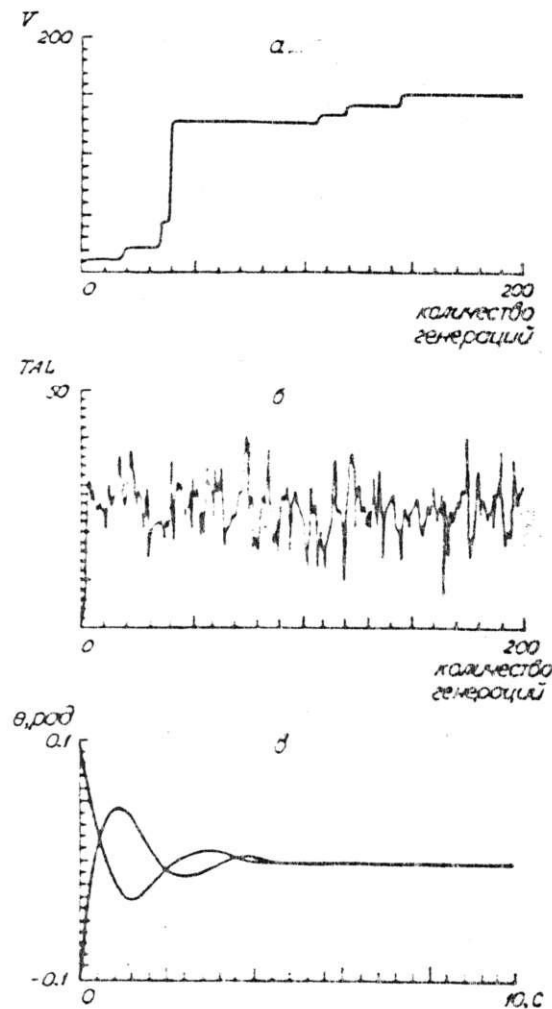


Рис. 15. Результаты моделирования процесса управления перевернутым маятником: а — значение коэффициента V для лучшей хромосомы; б — значение коэффициента соответствия; в — реакция перевернутого маятника для двух начальных значений для лучшей хромосомы при 200 поколениях

НС. На рис. 14 представлена структурная схема управления перевернутым маятником. Четыре переменные ($x, \dot{x}, \theta, \dot{\theta}$), описывающие состояние объекта управления являются входными переменными для 3 — уровневой нейронной сети, имеющей 25 весовых коэффициентов связи. Весовые коэффициенты представлены хромосомой (см. рис. 14). Управление $u = 5$, если выход объекта управления положителен и $u = -5$, если отрицателен, т. е. является нелинейным импульсным сигналом. Уравнение движения такого объекта управления приводилось в части III данной работы [59]. Эволюция каждой хромосомы описана в [218]. Значение величины V использовалось совместно с мерой пригодности и динамическим размахом оператора мутации. Процесс сходимости для 20 популяций приведен на рис. 15. Лучший результат поведения перевернутого маятника для двух начальных условий ($\theta = \pm 0.1$) приведен на рис. 15, в и иллюстрирует хорошие свойства процесса управления, реализованного только на интеллектуальной инструментари выбор сигнала.

Как видно из этого примера, ГА и НС представляют собой как бы две стороны одной медали, составляющей основу моделирующей системы. Их совместное

использование на этапе имитационного моделирования позволяет решить весьма сложную задачу по формированию необходимой управляющей системы, основные характеристики которой обеспечиваются не за счет усложнения моделей исполнительного уровня, а за счет привлечения более развитых средств интеллектуальной надстройки, предоставляемых возможностями современной информатизированной инструментальной отладочной среды ИНСАУ на этапе имитационного моделирования, в результате которого имеется возможность получения исчерпывающих данных, отражающих динамику проектируемой системы.

Заключение. В статье рассмотрены основные принципы формирования баз знаний интеллектуальных систем управления методами имитационного моделирования на основе совместного использования генетических алгоритмов и нейронных сетей. Разработанная методология имитационного моделирования является основой специального инструментального проектирования интеллектуальных управляющих систем (FZUP-методологией проектирования [39, 57]). Использование биологических аналогов в обработке больших массивов информации в поиске решений в слабоструктурированных средах в виде генетических алгоритмов и нейронных сетей позволяет использовать выбранные природой механизмы для создания эффективных средств проектирования без знаний интеллектуальных систем управления. Такой подход к созданию ИНСАУ одновременно позволяет воссоздать объективную форму проверки экспертных знаний и работать на всей области возможных решений. Нечеткие аналоги генетических алгоритмов и нейронных сетей дают возможность формирования соответствующих баз знаний интеллектуальных систем управления, работающих в условиях неполноты информации, и реализации принципа самоорганизации при неполной информации о внешней среде в таких системах.

СПИСОК ЛИТЕРАТУРЫ

1. Tools abound for incorporated fuzzy logic into today's designs // *Cogn. Rept.* 1992. V. 4. № 10.
2. Texas instruments describes details of experimental fuzzy chip // *Cogn. Rept.* 1993. V. 5. № 5.
3. Today purchases process control division of intermetrics // *Cogn. Rept.* 1993. V. 5. № 5.
4. Neue Impulse durch Fuzzy Logic: Technische Lösungen mit sprachlichen Regeln // *Techn. Rept.* 1993. V. 20. № 10.
5. Anderson C. Applying fuzzy logic in the real world // *Sensor.* 1992. V. 9. № 9.
6. Kahaner D. K. Advanced in the fuzzy theory and applications // *IEEE Micro.* 1981. V. 11. № 4.
7. Berardinis L. A. Clear thinking on fuzzy logic // *Mach. Des.* 1992. V. 64. № 8.
8. Warburton D. How to design fuzzy logic controllers // *Mach. Des.* 1992. V. 64. № 23.
9. Layne J. R., Passino K. M., Yarkovich S. Fuzzy learning control for Antiskid Braking Systems // *IEEE Trans. Contr. Syst. Technol.* 1993. V. 1. № 2.
10. Мацумото К. Переход от компактности к изяществу в нечеткой логике // *Эрэкутороникусу.* 1990. Т. 35. № 10.
11. Янагихара К. Система управления ORPX-Fuzzy / *Отомасен.* 1990. Т. 35. № 4.
12. Yager R. R. Implementing fuzzy logic controller using a neural network framework // *Fuzzy Sets and Systems.* 1992. V. 48. № 1.
13. Von Althrock C., Krause B., Zimmermann H.-J. Advanced fuzzy logic control of a model car in extreme situations // *Fuzzy Sets and Systems.* 1992. V. 48. № 1.

14. Fuzzy control and introduction to home appliances // *Techno Jap.* 1991. V. 24. № 9.
15. *De Yong M., Polson J., Moore R. et al.* Fuzzy and adaptive control simulations for a walking machine // *IEEE Contr. Syst. Mag.* 1992. V. 12. № 3.
16. *Takata K.* Research refines fuzzy logic technology, clarifies artificial intelligence use // *Data Commun. Process.* 1991. V. 23. № 12.
17. *Lea R., Jani Y.* Design and performance comparison of fuzzy logic based tracking controllers // *AIAA Pap.* 1992. № 1596.
18. *Suzuki K.* New control method utilizing fuzzy logic // *Ebara Eng. Rev.* 1992. № 155.
19. *Tamura H., Kutsumi T., Hatono I.* Fuzzy control of an interconnected multi-reservoir power system combined with real-time prediction: Comparison with optimal regulator through simulation studies // *Syst. Contr. and Inf.* 1991. V. 35. № 10.
20. *Lei Y.* The design of a new fuzzy controller for PID tuning // *Process Autom. Instrum.* 1993. V. 14. № 3.
21. *Hirota K.* Fuzzy control. // *J. Robot. Soc. Jap.* 1991. V. 9. № 2.
22. *Hussu A.* Fuzzy controllers // *Strojn. Vestn.* 1992. V. 38. № 10—12.
23. *Cheng Y., Zhuang C., An X.* A general fuzzy logic controller and its application // *Acta Autom. Sin.* 1992. V. 18. № 6.
24. *Hamilton S. D., Bell R. D., Bull D. J.* Fuzzy logic controller for robotics // *Inst. Eng. Austral.* 1992. № 92/15.
25. *Maeda M., Murakami S.* A design for a fuzzy logic controller // *Inf. Sci.* 1989. V. 45. № 2.
26. *Isaka S., Sebald A. V.* An optimization approach for fuzzy controller design // *IEEE Trans. Syst., Man. and Cybernetics.* 1992. V. 22. № 6.
27. *Hsu Y.-Y., Cheng C.-H.* A fuzzy controller for generator excitation control // *IEEE Trans. Syst., Man. and Cybernetics.* 1993. V. 23. № 2.
28. *Han J.-Y., McMurray V.* Two-layer multiple variable fuzzy controller // *IEEE Trans. Syst., Man. and Cybernetics.* 1993. V. 23. № 1.
29. *Watanabe K.* Mean-value-based functional reasoning and its application to the design of fuzzy controllers // *Repts Fac. Sci. Eng. Saga Univ.* 1993. V. 22. № 1.
30. *Liu T. S., Wu L. C.* A model for rider-motocycle systems using fuzzy control // *IEEE Trans. Syst., Man. and Cybernetics.* 1993. V. 23. № 1.
31. *Hayashi K., Ohtsubo A., Muta I. et al.* Simulator for fuzzy control with indirect inference and fuzzy PID method // *Repts Fac. Sci. Eng. Saga Univ.* 1993. V. 22. № 1.
32. *Abdelnour G., Cheung J. Y., Chung C.-H. et al.* Application of describing functions in the transient response analysis of a three-term fuzzy controller // *IEEE Trans. Syst., Man. and Cybernetics.* 1993. V. 23. № 2.
33. *Uflyanov S. V.* Fuzzy models of intelligent control systems: Theoretical and applied aspects // *Sov. J. Comp. and Systems Sci.* 1992. V. 30. № 4.
34. *Захаров В. Н., Ульянов С. В.* Нечеткие модели интеллектуальных промышленных регуляторов и систем управления. I // *Изв. АН СССР. Техн. кибернетика.* 1992. № 5.
35. *Терано Т., Асаи К., Сугено М.* Прикладные нечеткие системы. М.: Мир, 1993.
36. *Linkens D. A., Abbod M. F.* Self-organising fuzzy logic control and the selection of its scaling factors // *Trans. Inst. MC.* 1992. V. 14. № 3.
37. *Linkens D. A., Abbod M. F.* Supervisory intelligent control using a fuzzy logic hierarchy // *Trans. Inst. Meas. and Contr.* 1993. V. 15. № 3.
38. *Nagataki K., Kotsuji C., Yahiro M.* A scheme and operation results of a road tunnel ventilation control using hybrid expert system technology // *Hitachi Rev.* 1992. V. 41. № 1.

39. Захаров В. Н., Ульянов С. В. Нечеткие модели интеллектуальных промышленных регуляторов и систем управления. III. Методология проектирования // Изв. РАН. Техн. кибернетика. 1993. № 5.
40. Кацурагава М. Что может контроллер на нечеткой логике? // Эрокуторо-никусу. 1990. Т. 35. № 10.
41. Sano H., Morimoto Sh., Oba K. et al. Fuzzy controller with realtime tuning function // Yasukawa Techn. Rev. 1992. V. 56. № 4.
42. Fuzzy servo controller // Mitsubishi Heavy Ind. Techn. Rev. 1992. V. 29. № 3.
43. Wegmann H. Fuzzy control with SIMATIC S5 // Eng. and Automation. 1992. V. 14. № 6.
44. Fuzzy — logic para miniautomatas SIMATIC // Met. y Elec. 1993. V. 57. № 656.
45. Захаров В. Н., Ульянов С. В. Принципы построения моделей интеллектуальных машин и когнитивных систем управления // Проблемы обработки информации в робототехнических системах. М.: ИФТП РАН, 1993.
46. Schlag M., Chan P. K., Kong J. Empirical evaluation of multilevel logic minimization tools for a lookup-table-based field-programmable gate array technology // IEEE Trans. Comput. Aid. Des. Integr. Circuits and Syst. 1993. V. 12. № 5.
47. Williams T. ASIC cora brings fuzzy logic to semiconductors // Comput. Des. 1992. V. 31. № 6.
48. MacLeod A. When maths fails, fuzzy logic rules // New Electron. 1992. V. 25. № 4.
49. Wang J. Z., Yang H. A class of intelligent control system // Bull. Appl. Math. 1992. V. 63. № 6.
50. King K.-H. The status and future development of fuzzy logic control system // Nat. Contr. Publ. Inst. Eng. Austral. 1992. № 92/15.
51. Terano T. Human friendly system through fuzzy engineering // J. Soc. Instrum. and Contr. Eng. 1992. V. 31. № 1.
52. Diamond J., McLeod R. D., Pedrycz W. A fuzzy cogitive structure: Foundations, applications and ULSI implementation // Fuzzy Sets and Systems. 1992. V. 47. № 1.
53. Zhou H.-h. A practical intelligent technology of self-searching for fuzzy parameter optimization // Process Autom. Instrum. 1993. V. 14. № 3.
54. Bonnissonne P. P., Chiang K. H. Fuzzy logic controllers: From development to deployment // IEEE Int. Conf. Neural Networks. San Francisco, Calif., March 28 — Apr. 1, 1993. V. 2.
55. Zadeh L. A. Foreword // J. Intell. Fuzzy Systems. 1993. V. 1. № 1.
56. Захаров В. Н., Ульянов С. В. Нечеткие модели интеллектуальных промышленных регуляторов и систем управления. II. Эволюция и принципы построения // Изв. РАН. Техн. кибернетика. 1993. № 4.
57. Ulyanov S. V., Zhakharov V. N. Towards design of AI control systems // Proc. East-West Conf. Art. Intelligence: From Theory to Practice. EWAIC'93. Moscow, Sept. 7—9, 1993.
58. Kosko B. Neural networks and fuzzy systems: A dynamical approach to machine intelligence. Engle Wood Cliffs. N. Y.: Prentice-Hall. 1992.
59. Werbos P. Neurocontrol and fuzzy logic — connections and designs // Int. J. Approx. Reasoning. 1992. V. 6. № 2.
60. Fukuda T., Shibata T. Theory and applications of neural networks for industrial control systems // IEEE Trans. Industrial Electronics. 1992. V. 39. № 6.
61. Dote Y. Fuzzy and neural network controller // Proc. 16th Annu. Conf. IEEE Ind. Electron. Soc. Pacific Grove, Calif., Nov. 27—30, 1990. V. 2.
62. Yager R. R. Implementing fuzzy logic controllers using a neural network framework // Fuzzy Sets and Systems. 1992. V. 48. № 1.

63. *Bezdek J. C.* Fuzzy models — What are they, and why? // IEEE Trans. Fuzzy Systems. 1993. V. 1. № 1.
64. *Hayashi I.* Fusion method between fuzzy control and neural networks // *Joho Shori*. 1993. V. 34. № 1.
65. *Karr C. L.* Genetic algorithms for fuzzy controllers // *AI Expert*. 1991. V. 6. № 1.
66. *Karr C. L., Harper T. R.* Genetic algorithms in adaptive fuzzy control // Proc. Conf. NAFIPS'92. Puerto Vallarta, Mexico, December 15—17, 1992. V. 1.
67. *Shehaden H., Lea R.* A genetic algorithms approach for altering the membership functions in fuzzy logic controller // Proc. Conf. NAFIPS'92. Puerto Vallarta, Mexico, December 15—17, 1992. V. 1.
68. *Karr C. L., Gentry E.* Fuzzy control of pH using genetic algorithms // IEEE Trans. Fuzzy Systems. 1993. V. 1. № 1.
69. *Steinberg M.* Potential role of neural networks and fuzzy logic in flight control design and development // AIAA Aerospace Design Conf. Irvine, February 3—6, 1992.
70. *Петров Б. Н., Уланов Г. М., Ульянов С. В.* Теория моделей в процессах управления: Информационный и термодинамический аспекты. М.: Наука, 1978.
71. *Unemi T.* Applications of GA to control. // *J. Soc. Instrum. and Control Eng.* 1993. V. 32. № 1.
72. *Anderson B. D. O., Liu Y.* Controller reduction: Concept and approaches // IEEE Trans. Autom. Control. 1989. V. 34. № 8.
73. *Anderson B. D. O.* Controller design: Moving from theory to practice // IEEE Contr. Syst. Mag. 1993. V. 13. № 4.
74. *Kokotovic P. V.* The joy of feedback: nonlinear and adaptive // IEEE Contr. Syst. Mag. 1992. V. 12. № 3.
75. *Matko D.* Adaptive or robust control? Robust adaptive control! // *Elektrotechn. Vestn.* 1993. V. 60. № 2, 3.
76. *Denker A., Kaynak O.* SLIMSOC: A merger of sliding mode and self-organization controllers with application to industrial robots // *Elektrotechn. Vestn.* 1993. V. 60. № 2, 3.
77. *VanderLinden G.-W., Lambrechts P. F.* H_∞ control of an experimental inverted pendulum with dry friction // IEEE Contr. Syst. Mag. 1993. V. 13. № 4.
78. *Juricic D., Petrovic J.* Some issues related to the modelling and control of nonlinear unstable plants — a case study // *Elektrotechn. Vestn.* 1993. V. 60. № 2, 3.
79. *Mahmoud M., Eid S. Z., Abou-Elsoad A. A.* A real-time expert control system for dynamical processes // IEEE Trans. Syst., Man, and Cybern. 1989. V. 19. № 5.
80. *Li J., Xi Y.* Dynamic matrix control with expert knowledge-based multi-layered heuristic self-tuning // *Inf. and Contr.* 1992. V. 21. № 5.
81. *Kokate P. Y., Das M., Loh R. N. K.* Rule-based control of slowly varying systems using compensator segmentation determined by simulated annealing // IEEE Trans. on Industr. Electron. 1992. V. 39. № 4.
82. *Liang J., Yang Y., Lu Y.* DMES: real-time expert control system // *Inf. and Contr.* 1993. V. 22. № 2.
83. *Hu Z., Jiang W.* Expert control for distillation columns // *Inf. and Contr.* 1993. V. 22. № 2.
84. *Wang Y.-ch.* Some comments on the development of expert system and intelligent controllers // *Process Autom. Instrum.* 1993. V. 14. № 1.
85. *Chi W., Gao D.* On Intelligent design of control systems // *Contr. Theory and Appl.* 1993. V. 10. № 3.
86. *Harris C. J., Moore C.* Real time fuzzy based self-learning predictors and controllers // Proc. 11th Trienn. World Congr. IFAC (Tallinn, Aug. 13—17, 1990). Oxford, 1991. V. 1.

87. *Vander Rhee F., Van Nauta Lemke, Dijkman J. G.* Knowledge based fuzzy modelling of systems // Proc. 11th Trienn. World Congr. IFAC (Tallinn, Aug. 13—17, 1990). Oxford, 1991. V. 1.
88. *Ruckly J. J., Ying H.* Fuzzy controller theory: Limit theorems for linear fuzzy control rules // *Automatica*. 1989. V. 25. № 3.
89. *Matia F., Jimenez A., Galan R. et al.* Fuzzy controllers: Lifting the linear-nonlinear frontier // *Fuzzy Sets and Systems*. 1992. V. 52. № 2.
90. *Hayashi K., Ohtsubo A., Muta I. et al.* Simulator for fuzzy control with indirect inference and fuzzy PID method // *Repts Fac. Sci. Eng. Saga Univ.* 1993. V. 22. № 1.
91. *Yager R. R., Filev D. P.* SLIDE: A simple adaptive defuzzification method // *IEEE Trans. Fuzzy Systems*. 1993. V. 1. № 1.
92. *Figueiredo M., Gomide F., Rocha A. et al.* Comparison of Yager's level set methods for fuzzy logic control with Mamdani's and Larsen's methods // *IEEE Trans. Fuzzy Systems*. 1993. V. 1. № 2.
93. *Алиев Р. А., Церковный А. Э., Мамедова Г. А.* Управление производством при нечеткой исходной информации. М.: Энергоатомиздат, 1991.
94. *Abdelnour G. M., Chang C.-H., Huang F.-H. et al.* Design of fuzzy controller using input and output mapping factors // *IEEE Trans. System, Man, and Cybernetics*. 1991. V. 21. № 5.
95. *Ulianov S. V.* Fuzzy control motion of the mobile robots for horizontal and vertical displacement in unstructured environments // *Proc. World Congress on Expert System. Lisbon / Estoril Portugal. January 10—14, 1994. V. 1.*
96. *Raju G. V. S., Zhou J., Kisner R. A.* Hierarchical fuzzy control // *Int. J. Control*. 1991. V. 54. № 5.
97. *Batur C., Kasparian V.* Model based fuzzy control // *Math. Comput. Modell.* 1991. V. 15. № 12.
98. *Yager R. R.* Fuzzy logic controller structures // *SPIE Intell. Robots and Computer Vision IX: Algorithms and Techniques*. 1990. V. 1381.
99. *Kwok D. P., Wang P., Li C. K.* A combined fuzzy and classical PID control // *Microprocess. and Microprogramm.* 1991. U. 3. № 1—5.
100. *Ding J.* PID control algorithm for on-line autotuning of parameters // *J. China Text. Univ.* 1989. V. 15. № 3.
101. *He M., Lu Y.* Hybrid intelligent controller // *Inf. and Contr.* 1988. № 4.
102. *Saito Y., Ishida T.* Fuzzy PID hybrid control — an application to burner control // *Proc. Fuzzy Syst. and Neural Networks, Iizuka, Japan, July 20—24, 1990.*
103. *Jang S., Araki M.* Mathematical analysis of fuzzy control system and on possibility of industrial applications // *Trans. Soc. Instrum. and Cont. Eng.* 1990. V. 26. № 1.
104. *Lin C. E., Sheu Y.-R.* A hybrid-control approach for pendulum-car control // *IEEE Trans. on Industrial Electronics*. 1992. V. 39. № 3.
105. *Saito T., Togawa K.* Controls of inverted pendulum — By the technique using the analog control elements // *Res. Repts. Nagaoka Techn. Coll.* 1991. V. 27. № 2.
106. *Kawabe H., Saiki H., Yoshida K.* Stabilizing control of an inverted pendulum using USS theory: Comparison on robustness between a sliding mode control with a single switching gain and with a 4 switching gains // *Res. Bull. Hiroshima. Inst. Technol.* 1993. V. 27. № 1.
107. *Hwang G.-C., Lin S.-C.* A stability approach to fuzzy control design for nonlinear systems // *Fuzzy Sets and Systems*, 1992. V. 48. № 3.
108. *Li Z.-S., Takahashi T., Yu K.-H. et al.* Control of a pendulum under torque limitation (swinging-up control by intelligent controller) // *IEEE Trans. Tap. Soc. Nech. Eng. C*. 1992. V. 58. № 3.
109. *Xia Y., Inooka H.* Application of tree search to the swinging control of a pendulum // *IEEE Trans. Jap. Soc. Mech. Eng. C*. 1992. V. 22. № 5.

110. *Wieland A. P.* Evolving neural controllers for unstable system // Int. Jt Conf. Neur. Networks. Seattle, Wash., July 8—12, 1991. V. 2.
111. *Hao J., Tan S., Vandewalle J.* A rule-based neural controller for inverted pendulum systems // IEEE Int. Conf. Neur. Networks. San Francisco, Calif., March 28 — April 1, 1993. V. 1.
112. *Ying X., Zeng N.* A controller implemented by recording the fuzzy rules by BP neural networks // Acta Autom. Sin. 1991. V. 17. № 1.
113. *Wang Li-X., Mendel J. M.* Fuzzy basis functions, universal approximations, and orthogonal least-squares learning // IEEE Trans. Neural Networks. 1992. V. 3. № 5.
114. *Karr C. Z.* Design of an adaptive fuzzy logic controller using a genetic algorithm // Proc. 4th Int. Conf. on Genetic Algorithms. San Diego, Calif., 17—21 June, 1991.
115. *Thrft P.* Fuzzy logic synthesis with genetic algorithm // Proc. 4th Int. Conf. on Genetic Algorithms. San Diego, Calif., 17—21 June, 1991.
116. *KrishnaKumar K.* Genetic algorithms: An introduction and an overview of their capabilities // AIAA Guid. Navig. and Confr. Cont. Hilton Head Island, S. C., Aug. 10—12, 1992. Pt 2.
117. *Shibata T., Fukuda T.* Genetic algorithm with fuzzy critic for robotic motion planning // ICAR. Tokyo, Nov. 3—7, 1993.
118. *Linkens D. A., Wie J.* Neural network-based approximate reasoning // Int. J. Control. 1992. V. 56. № 4.
119. *Linkens D. A., Wie J.* Fuzzified RBF network-based learning control: structure and self-construction // IEEE Int. Conf. Neural Networks. San Francisco, Calif., March 28 — Apr. 1. 1993. V. 2.
120. *Kawase S., Yanagishare N.* Representation of dynamical system by fuzzy reasoning // Trans. Inst. Electron., Inf. and Commun. Eng. A. 1992. V. 75. № 4.
121. *Петров Б. Н., Ульянов Г. М., Ульянов С. В.* Сложность конечных объектов и информационная теория управления // Итоги науки и техники. Сер. Техн. кибернетика. М.: ВИНТИ РАН. 1979. Т. 11.
122. *Kosko B.* Fuzzy systems as universal approximators // Proc. 1-st IEEE Int. Conf. Fuzzy Systems. FUZZ-IEEE FUZ'92. San Diego, CA, March, 1992.
123. *Wang L-X.* Fuzzy systems are universal approximators // Proc. 1-st IEEE Int. Conf. Fuzzy Systems. FUZZ-IEEE FUX'92. San Diego, CA, March, 1992.
124. *Wang L-X., Mandel J. M.* Generating fuzzy rules by learning from examples // IEEE Trans. Syst., Man. and Cybern. 1992. V. 22. № 6.
125. *Cappelle B., de Cooman G., Kerre E. E. et al.* Intelligent reduction methods based upon fuzzy set theory as a helpful tool for controlling industrial plants // Fuzzy Sets and Systems. 1993. V. 50. № 2.
126. *Sugeno M., Yasukawa T.* A fuzzy-logic-based approach to qualitative modelling // IEEE Trans. Fuzzy Syst. 1993. V. 1. № 1.
127. *Pedrycz W.* Relevancy of fuzzy models // Inf. Sci. 1990. V. 52. № 3.
128. *Stefanov S. Z.* General theory of intelligent systems // Int. J. Cybern. and Systems. 1992. V. 23. № 5.
129. *Gradetsky V. G., Ulyanov S. V., Yamafuji K.* Knowledge-based fuzzy control systems: Principles of software design and its applications // Proc. of JCKBSE'94: Japan-CIS Symposium on Knowledge-Based Software Eng. 94. Pereslavl-Zaleski, Russia, May 10—13, 1994.
130. *Yager R. R., Larsen H. L.* On discovering potential inconsistencies in validating uncertain knowledge bases by reflecting on the input // IEEE Trans. Syst., Man. and Cybern. 1991. V. 21. № 4.
131. *Yager R. R.* Deductive approximate reasoning systems // IEEE Trans. Knowledge and Data Eng. 1991. V. 3. № 4.

132. *Mittelstadt P.* Relativistic quantum logic // *Int. J. Theoret. Phys.* 1983. V. 22. № 4.
133. *Mielnik B.* Motion and Form // *Current Issues in Quantum Logic* / Eds Bellrametti E. G., Fraasen B. C. (E. Majoruna Intern. Sci. Ser.: Phys. Sci.). N. Y.: Plenum Press, 1981. V. 8.
134. *Гриб А. А., Запартин Р. Р.* Квантовая логика — проблемы и перспективы // *Семиотика и информатика*. М.: ВИНТИ РАН. 1989. Т. 29.
135. *Самарский А. А., Галактионов В. А., Курдюмов С. П.* Режимы с обострением в задачах для квазилинейных параметрических уравнений. М.: Наука, 1987.
136. *Mittelstadt P.* Quantum logic. Holland, Dordrecht: D. Redel Publ. Co., 1978.
137. *Ptak P., Pulmannova S.* Kvantove logiky. Bratislava, VEGA, 1989.
138. *Beltrametti E. G., Gassinelli G.* The logic of quantum mechanics. *Encyclopedic of Mathematics and its Applications*. Addison-Wesley Publ. Co., 1981. V. 15.
139. *Budajski S.* Languages of similirity // *J. Phil. Logic*. 1983. V. 12. № 1; Probability implication in the logic of classical and quantum mechanics // *Ibid.* 1978. V. 7. № 2.
140. *Dalla Chiara M. L., Toraldo di Francia G.* Formal analysis of physical theories // *Algebraic Formulation in Quantum Mechanics*. Dordrecht, Holland, 1979.
141. *Морозов Б. З.* Формальные системы, возникающие при анализе физических теорий // *Докл. АН СССР*. 1971. Т. 198. № 5.
142. *Ахиезер А. И., Половин Р. В.* Почему невозможно ввести в квантовую механику скрытые параметры // *УФН*. 1972. Т. 107. Вып. 3.
143. *Ульянов С. В.* Модели квантово-релятивистских нечетких логик в интеллектуальных системах // *Докл. 2-й Всесоюз. конф. «Искусственный интеллект — '90»*. М.: ВЦ АН СССР. 1990. Т. 2.
144. *Dalla Chiara M. L.* The relevance of quantum logic in the domain of non-classical logics // *Logic, Methodology and Philosophy of Science* / Eds B. Marcus et al. Elsevier Publ., 1986. V. 7.
145. *Гольденблат И. И., Ульянов С. В.* Введение в теорию относительности и ее применения к новой технике. М.: Наука, 1979.
146. *Burghardt F. J.* Modalities and quantum mechanics // *Int. J. Theoret. Phys.* 1984. V. 23. № 12; Modal quantum logic and its dialogic foundation // *Ibid.* 1980. V. 19. № 11.
147. *Hertz J., Krough A., Palmer R. G.* Introduction to the theory of neural computation. Addison-Wesley, Reading, MA, 1991.
148. *Muller B., Reinhart I.* Neural networks-an Introduction. Berlin: Springer Verlag, 1991.
149. *Werbos P. I.* New control and related techniques. *Hand Book of neural computing applications*. N. Y.: Acad. Press, 1990.
150. *Grossberg S., Kuperstein M.* Neural dynamic of adaptive sensor-motor control. Pergamon Press, Elseford. N. Y., 1989.
151. *Kohonen T.* Self-organization and associative memory. Springer, Verlag, Berlin, 1988.
152. *Schalkoff R.* Pattern recognition: Statistical Structural and Neural Approaches. N. Y.: Wiley & Sons, 1992.
153. *Mead C.* Analog ULSI and Neural Systems, Addison-Wesley, MA, 1989.
154. Special Issue on Neural Networks in Control Systems // *IEEE Contr. Syst.*, May, 1992. V. 15. № 2.
155. *Zarada J. M.* Introduction to Artificial Neural Systems. West. St. Paul, MN, 1992.
156. *Hewetson M. et al.* Pygmalion neurocomputing tutorials. Univ. College. London, 1991.
157. *Amit D.* Modelling brain function. Cubridge: Cambridge Univ. Press, 1985.

158. Charabarti B. K., Dasgupta P. K. Modelling neural networks // *Physica*. 1992. V. 186. A. № 1.
159. Gupta M. M., Rao D. H. Dynamic neural units with applications to the control of unknown nonlinear systems // *J. of Intelligent and Fuzzy Systems*. 1993. V. 1. № 1.
160. Tusar M., Zupan J., Gasteiger J. Neural networks and modelling in chemistry // *J. Chim. Phys. et Phys.-Chim. Biol.* 1992. V. 89. № 7, 8.
161. Wang de L. Pattern recognition: Neural networks in perspective // *IEEE Expert*. 1993. V. 8. № 4.
162. Dugupta C. Neural network of associative memory beyond the Hopfield model // *Physica*. 1992. V. 186 A. № 2.
163. Rumelhart D. E., Hinton G. E., Williams R. J. Learning internal representations by error propagation // *Parallel Distributed Processing* / Eds D. E. Rumelhart and J. L. McClelland. MIT Press, Cambridge, Mass., 1986. V. 1.
164. Rosenblatt F. Principles of neural dynamics. Spartan. N. Y., 1962.
165. Miller W. T., Sutton R. S., Werbos P. T. Neural networks for control. Cambridge: MIT Press, 1990.
166. Swiniarski R. W. Novel neural network based self-tuning PID controller wich uses pattern recognition technique // *American Control Conf. San Diego, Calif.* May 12—14, 1990.
167. Yamada T., Takahashi K., Yasue N. et al. Remarks on a learning type self-tuning neural network controller // *ICAR'93. Japan*. 1993.
168. Akhyar S., Omatu S. Neuromorphic self-tuning PID controller // *IEEE Int. Conf. Meural Networks*. San Francisco, Calif., March 28 — Apr. 1, 1993. V. 1.
169. Scott G. M., Shalvlik J. W., Ray W. H. Refining PID controllers using neural networks // *Neural Comput.* 1992. V. 4. № 5.
170. Ekond D. U., Abd-el-Barr M. H., Wood H. C. Design of fault tolerant neurocontrollers using immunization technique // *IEEE Int. Conf. Neur. Networks*. San Francisco, Calif., March 28 — April 1, 1993. V. 3.
171. Narendra K., Parthasarathy K. Identification and control of dynamical system using neural networks // *IEEE Trans. Neural Networks*. 1992. V. 1. № 1; Narendra K. S., Mukhopadhyay S. Intelligent control using neural networks // *IEEE Contr. Syst. Mag.* 1992. V. 12. № 2; Levin A. U., Narendra K. S. Control of nonlinear dynamical systems using neural networks: Controllability and stabilization // *IEEE Trans. Neural Networks*. 1993. V. 4. № 2.
172. Kuchewski J. G., Hui S., Zak S. Application of feedforward neural network for dynamical system identification and control // *IEEE Trans. Cout. Syst. Technol.* 1993. V. 1. № 1.
173. Chessari C. J., Barton G. W. Model based control using neural network models // *Inst. Eng. Austral.* 1992. № 92/15.
174. Schalchter D. B. Applications of neural network to control system // *AIAA Guid. Navig. and Confr. Cont.* Hilton Head Island. S. C., Aug. 10—12, 1992. Pt 1.
175. Samad Y. Neurocontrol: Concepts and applications // *IEEE Cont. Syst., Man. and Cybern.* Chicago. Oct. 18—21, 1992. V. 2.
176. Luebbers P. G., Pandya A. S. Neural networks for process control // *IEEE Cont. Syst., Man. and Cybern.* Chicago. Oct. 18—21, 1992. V. 2.
177. Yabuta T., Yamada Y. Neural network controller characteristics with regard for adaptive control // *IEEE Trans. Syst., Man. and Cybern.* 1992. V. 22. № 1.
178. Levin E. Hidden control neural architecture modelling of nonlinear time varying systems and its application // *IEEE Trans. Neural Networks*. 1993. V. 4. № 1.
179. Van Landingham H. F., Bingulac S., Tran M. A comparison of conventional and neural network approaches to system identification // *Control Theory and Technol.* 1993. V. 9. № 1.

180. Wang H., Lee T. T., Graver W. A. A neuromorphic controller for a three-link hiped robot // IEEE Trans. Syst., Man, and Cybern. 1992. V. 22. № 1.
181. Sartoric M., Antsaklis P. J. Implementations of learning control system using neural networks // IEEE Contr. Syst. Mag. 1992. V. 12. № 2.
182. Chu S. R., Shoureshi R. Applications of neural networks in learning of dynamical systems // IEEE Contr. Syst. Mag. 1992. V. 12. № 2.
183. Calise A., Kim B. S., Kam M. et al. Neural networks for feedback linearization in aircraft control // AIAA Guid. Navig. and Confr. Cont. Hilton Head Island, S. C., Aug. 10—12, 1992. Pt 1.
184. Di Girolamo R. Flight control law systems using neural network theory // AIAA Guid. Navig. and Confr. Cont. Hilton Head Island, S. C., Aug. 10—12, 1992. Pt 1.
185. Cybenko G. Approximation by superposition of sigmoidal function // Mathematics of Control. Signals and Systems. 1989. V. 2. № 1.
186. Funahashi K. I. On the approximate realization of continuous mapping by neural networks // Neural Networks. 1989. V. 2. № 2.
187. Hornik K., Stinchcombe M., White H. Multilayer feedforward network are universal approximators // Neural Networks. 1989. V. 2. № 3.
188. Barron A. R. Universal approximation bounds for superpositions of a sigmoidal function // IEEE Trans. Inf. Theory. 1993. V. 39. № 3.
189. Broomhead D. S., Lowe D. Multivariable functional interpolation and adaptive networks // Complex System. 1988. V. 2. № 3.
190. Omal M., Fujioka R., Ishibuchi H. et al. Learning algorithms of neural networks for interval-valued data // Bull. Univ. Osaka Prefect. A. 1991. V. 40. № 1.
191. Alefeld G., Herzberger J. Introduction to interval computation. N. Y.: Acad. Press, 1983.
192. Carpenter G. A., Grossberg S. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architecture // Neural Networks. 1990. V. 3. № 1.
193. Von der Malsburg C. Pattern recognition by labeled graph matching // Neural Networks. 1988. V. 1. № 1.
194. Lutz O., Dengel A. A comparison of neural net simulator // IEEE Expert. 1993. V. 8. № 4.
195. Mead C. Analog ULSI and neural systems. Addison-Wesley, 1989.
196. Акаэма Н. Техническое обеспечение нейросетей // Оперосндази Рисатси. 1992. Т. 37. № 7.
197. Horio Y., Wasaki H. Realization of analog neural chips: learning rule // Annu. Rept. Res. Inst. Technol.: Tokyo Denki Univ. 1991. № 11.
198. Eberhardt S. P., Tawel R., Brown T. X. et al. Analog ULSI neural networks: Implementation issues and examples in optimization and supervised learning // IEEE Trans. Ind. Electron. 1992. V. 39. № 6.
199. Zerbe K. Die Neuro-Chips kommen // CI-Mag. Comput. Techn. 1991. № 5.
200. Brauch J., Tam S. M., Holler M. et al. Analog ULSI neural networks for impact signal processing // IEEE Micro. 1992. V. 12. № 6.
201. Yokoi H. GH-Folthret as fundamental element for neural computer // Trans. Int. Electron., Int. and Commun. Eng. D2. 1993. V. 76. № 3.
202. Kyuma K., Lange E., Nitta Y. Optical neuro-device // Optoelectronics: Devices and Technology. 1993. V. 8. № 1.
203. Yonezu H., Tsuji K. Integrated optoelectronic neuro-device // Optoelectronics: Devices and Technology. 1993. V. 8. № 1.
204. Holland J. Adaptation in neural and artificial systems. Michigan: University of Michigan Press, 1975.
205. Goldberg D. E. Genetic algorithm in search, optimization, and machine learning. Addison-Wesley Publ. Cong., 1989.

206. *Davis L.* Handbook of genetic algorithms. Van Nostrand Reinhold, N. Y., 1991.
207. *Michalewicz Z.* Genetic algorithm + Data Structures = Evolution programs. N. Y.: Springer-Verlag, 1992.
208. *Koza J.* Genetic programming. Cambridge, Mass: MIT Press, 1992.
209. *Schwefer H.-P., Maenner R.* Parallel problem solving from nature. Lect. Not. in Comput. Sci. N. Y.: Springer-Verlag, 1991. V. 491.
210. *Rawlins G. J. E.* Foundations of genetic algorithms. Morgan Kaufmann, San Mateo, Calif., 1991.
211. *Whitley D.* Foundations of genetic algorithms. 2. Morgan Kaufman, San Mateo, Calif., 1991.
212. *Dorigo M., Schnepf V.* Genetic-based machine learning and behavior-based robotics: A new synthesis // IEEE Trans. Syst., Man. and Cybern. 1993. V. 23. № 1.
213. *Grefenstette J. J.* Genetic algorithms: Introductions // IEEE Expert. 1993. V. 8. № 5.
214. *De Jong K. A.* An analysis of the behaviour of genetic adaptive systems. Dissert. Abstr. Intern. 1975. V. 41. № 9.
215. *Grefenstette J. J.* Optimization of control parameters for genetic algorithm // IEEE Trans. Syst., Man. and Cybern. 1986. V. 16. № 1.
216. *Goldberg D. E., Deb K., Korb B.* Messy genetic algorithms: Motivation, analysis, and first results // Complex Systems. 1989. V. 3. № 3.
217. *Goldberg D. E., Deb K., Thierenes D.* Toward a better understanding of mixing in genetic algorithms // J. Soc. Instrum. and Contr. Eng. 1993. V. 32. № 1.
218. *Ichikawa Y., Ishii Y.* Retaining diversity of genetic algorithms for multivariable optimization and neural network learning // IEEE Int. Cont. Neural Networks. San Francisco, Calif., March 28 — Apr. 1, 1993. V. 2.

Москва,
ВЦ РАН,
АОЗТ «Робоцентр»

Поступила в редакцию
15.XII.1994