

Харьковский государственный политехнический университет

на правах рукописи

Махотило Константин Владимирович

УДК 519.7

**РАЗРАБОТКА МЕТОДИК ЭВОЛЮЦИОННОГО
СИНТЕЗА НЕЙРОСЕТЕВЫХ КОМПОНЕНТОВ
СИСТЕМ УПРАВЛЕНИЯ**

05.13.06 — Автоматизированные системы управления и
прогрессивные информационные технологии

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель
Сергеев Сергей Александрович
кандидат технических наук,
старший научный сотрудник

Харьков — 1998

СОДЕРЖАНИЕ

Список условных обозначений и сокращений	5
Введение	9
ГЛАВА 1 Искусственные нейронные сети — новая парадигма в управлении	15
1.1. Этапы развития искусственных нейронных сетей.....	17
1.2. Модель нейрона	20
1.3. Архитектура искусственных нейронных сетей.....	26
1.4. Обучение искусственных нейронных сетей.....	31
1.5. Свойства искусственных нейронных сетей.....	36
1.5.1. Свойство обобщения.	36
1.5.2. Свойство универсальной аппроксимации.	40
1.6. Применение искусственных нейронных сетей для решения задач управления.....	43
1.6.1. Идентификация.	47
1.6.2. Управление.....	50
1.7. Постановка задачи на исследование	56
ГЛАВА 2 Разработка методики синтеза нейроэмуляторов динамических объектов	57
2.1. РБФ-сети.....	59
2.2. Генетический алгоритм.....	63
2.3. Синтез РБФ-сети с помощью генетического алгоритма.....	74
2.4. Синтез нейроэмулятора динамического объекта.....	80
2.5. Идентификация динамического объекта по зашумленным данным	88
2.6. Выводы	95
ГЛАВА 3 Разработка методики синтеза нейросетевых систем управления.....	97

3.1. Использование генетического алгоритма для обучения нейроконтроллеров	99
3.2. Синтез нейросетевой системы управления с ПИД- нейроконтроллером	103
3.3. Нейросетевая система управления с заданными частотными характеристиками	111
3.4. Синтез дискретной нейросетевой системы управления.....	118
3.5. Выводы	126
ГЛАВА 4 Исследование свойств нейросетевых систем управления...	128
4.1. Синтез нейросетевой системы управления нелинейным объектом	130
4.1.1. Нейросетевая система управления на базе нейроконтроллера....	130
4.1.2. Синтез ПИД-нейроконтроллера для управления температурой пара на выходе котла ТГМП-204.	138
4.2. Синтез нейросетевой системы управления по нейроэмулятору объекта управления	145
4.3. Анализ параметрической и структурной чувствительности нейросетевой системы управления	154
4.4. Выводы	163
Заключение	166
Список использованных источников	169

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

A — особь

χ — аргумент целевой функции, вектор координат поисковой точки

χ^* — глобальное решение задачи оптимизации

X — область поиска

X_r — прямоугольная область поиска генетического алгоритма

Δ — шаг предсказания нейроэмулятора

ε_σ — минимальная ширина окна радиально-базисной функции

ε_φ — минимально допустимое значение аргумента радиально-базисной активационной функции

ε_c — минимально допустимое расстояние между двумя шаблонами в тренировочном наборе, радиус сжатия тренировочного набора шаблонов

ε_o — ошибка оценивания нейронной сети (рассчитывается с использованием данных не участвовавших в обучении)

γ_i — нижняя граница области поиска по i -й координате

η_i — шаг дискретизации прямоугольной области поиска по i -й проекции

$\varphi(\cdot)$ — активационная функция нейрона

\mathfrak{I} — интерполяционная матрица

λ_i — длина i -й хромосомы

μ — размер популяции

ν — размерность поискового гиперкуба

Π — популяция

θ — ошибка измерения выходного сигнала идентифицируемого объекта

ρ — коэффициент давления отбора

σ — ширина окна радиально-базисной активационной функции
 σ — ширина окна радиально-базисной активационной функции нейрона
 Σ — тренировочный набор сигналов
 τ — длительность тренировочного сигнала
 Ξ — тренировочный набор шаблонов
 Ξ_r — тренировочная выборка шаблонов
 a — параметр активационной функции нейрона
 b, w_{0k} — сдвиг нейрона
 \mathbf{c} — вектор координат центра радиально-базисной активационной функции нейрона
 c — координата центра радиально-базисной активационной функции нейрона
 d — желаемый вход нейронной сети
 E — суммарная ошибка реакции сети на тренировочном наборе шаблонов
 $F(\cdot)$ — функциональное преобразование, осуществляемое нейронной сетью
 $f(\cdot)$ — целевая функция задачи оптимизации
 g — размер входного слоя сети (количество входных нейронов)
 h — размер скрытого слоя сети
 k_i — целочисленного значения i -й координаты
 l — размерность пространства поиска, количество хромосом в одинарном наборе
 l_1 — глубина задержки обратной связи по выходу объекта управления (нейроэмулятора)
 l_2 — глубина задержки по входу объекта управления (нейроэмулятора)
 M — размер тренировочного набора сигналов
 m — размерность выходного пространства динамической системы
 N — количество шаблонов в тренировочном наборе (размер набора)

n — порядок динамической системы
 $NC(\cdot)$ — преобразование, осуществляемое нейроконтроллером
 $NE(\cdot)$ — преобразование, осуществляемое нейроэмулятором
 N_m — максимальный размер тренировочной выборки
 $NN(\cdot)$ — преобразование, осуществляемое нейронной сетью
 p — размерность входного пространства динамической системы
 $P\{\cdot\}$ — объект управления
 $\hat{P}\{\cdot\}$ — идентификационная модель
 q — размер выходного слоя сети (количество выходных нейронов)
 r — риск, определяющий способность нейронной сети обобщать новые данные
 $R\{\cdot\}$ — эталонная модель
 r_e — эффективный радиус популяции
 s — аргумент активационной функции нейрона
 t — время
 T_m — максимальная глубина предсказания нейроэмулятора
 \mathbf{u} — вектор сигналов управления
 U — допустимое множество сигналов управления
 u — сигнал управления
 u_r — тренировочный сигнал
 w_{jk} — j -й синаптический вес k -го нейрона
 \mathbf{x} — вектор переменных состояния динамического объекта
 x — переменная состояния динамического объекта
 \mathbf{y} — вектор выходных сигналов
 $\hat{\mathbf{y}}$ — вектор выходных сигналов идентификационной модели
 $\tilde{\mathbf{y}}$ — вектор выходных сигналов эталонной модели
 y — выходной сигнал
 z — входной сигнал нейрона или нейронной сети

ГА — генетический алгоритм

МСП — многослойный перцептрон

НК — нейроконтроллер

НСУ — нейросетевая система управления

НЭ — нейроэмулятор

РБФ — радиально-базисная функция

РБФ-сеть — искусственная нейронная сеть с радиально-базисной активационной функцией нейронов скрытого слоя

СУ — система управления

ВВЕДЕНИЕ

Основой технического прогресса, достигнутого человечеством в XX веке, стала массовая автоматизация производства. За полвека системы автоматического управления превратились из примитивных разомкнутых систем в сложные системы с обратной связью. Новое поколение систем автоматического управления должно не просто автоматизировать труд, но полностью или частично заменить человека при решении технических задач, требующих принятия интеллектуальных решений. Для этого они должны приобрести свойства гибкости, адаптивности и интеллектуальности.

Наилучшее решение поставленной задачи можно получить в рамках нейрокибернетического подхода, заимствуя механизмы и принципы функционирования интеллекта человека, выработанные в ходе эволюции самой Природой. Одним из наиболее мощных инструментов создания интеллектуальных систем управления являются искусственные нейронные сети, моделирующие базовые механизмы обработки информации в мозге. Исследования нейросетевых систем управления начались в 70-х годах и продолжают проводиться, привлекая все большее количество научных сил.

Объект управления в нейросетевых системах рассматривается, как черный ящик, подвергающийся внешним воздействиям. Цель системы управления — определить в процессе функционирования закон управления, который обеспечит оптимальное поведение объекта. Полученные результаты показали, что искусственные нейронные сети представляют собой не просто новое направление в теории автоматического управления, а самостоятельную парадигму, получившую название нейроуправление. В рамках этой парадигмы общими для различных объек-

тов управления методами решаются задачи идентификации, синтеза систем управления, их анализа и аппаратной реализации.

Разработка нейросетевых систем управления связана с существенными трудностями, так как со времен перцептрона Розенблатта нейронные сети традиционно использовались только при распознавании образов и классификации. Для построения на базе искусственных нейронных сетей регуляторов и идентификационных моделей необходимо разработать новые методы выбора их структуры и параметров. Одним из перспективных подходов к решению этой задачи является использование методов эволюционного моделирования, а именно генетических алгоритмов, для обучения и структурной оптимизации нейронных сетей.

Министерство Украины по делам науки и технологий определило приоритетные направления развития науки и техники, в рамках которых идет формирование государственных научно-технических программ. Такими направлениями являются, в частности, перспективные информационные технологии, устройства комплексной автоматизации, системы связи. Задачи, решаемые в настоящей диссертационной работе, ставились в соответствии с положениями, предусмотренными государственными научно-техническими программами (ГНТП), сформированными в рамках этих направлений.

ГНТП «Современные информационные технологии в создании интегрированных производственных комплексов». Разработка и внедрение:

- методов, алгоритмов и инструментальных способов системного анализа и разрешения практических задач управления;
- систем, методов и способов комплексной автоматизации технологических процессов.

ГНТП «Новые технологические приемы поддержки и принятия решений. Инструментально-технологические программные комплексы». Разработка и внедрение:

- способов и систем оптимизации и поддержки принятия решений;
- базовых инструментальных способов интеллектуализации информационных технологий и систем.

ГНТП «Создание конкурентоспособных приспособлений имитационного моделирования сложных систем, повышение их надежности и эффективности». Разработка и внедрение:

- математических моделей, методов, алгоритмов и программного обеспечения определения надежности и эффективности сложных технических систем;
- инструментально-технологических приспособлений моделирования сложных процессов и систем.

Целью диссертационной работы является разработка методик эволюционного синтеза нейросетевых компонентов систем автоматического управления, основывающихся на применении генетических алгоритмов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Адаптировать классическую схему генетического алгоритма для синтеза на базе прямонаправленных искусственных нейронных сетей идентификационных моделей динамических объектов;
2. Оценить перспективы применения идеологии задержанного стимулируемого обучения для тренировки компонентов нейросетевых систем управления с помощью генетического алгоритма;
3. Определить достаточную систему типовых входных воздействий, используемых для обучения нейросетевых компонентов систем управления;

4. Оценить верхнюю границу размера скрытого слоя трехслойных нейронных сетей, используемых для решения задач идентификации и управления. Разработать методику выбора структуры нейросетевых идентификационных моделей и нейроконтроллеров;
5. Оценить параметрическую и структурную чувствительности синтезируемых с помощью генетического алгоритма нейроконтроллеров;
6. Разработать методику синтеза нейроконтроллера по нейросетевой модели динамического объекта управления.

При исследованиях применялись методы классической теории автоматического управления, а также методы нейрокибернетики и эволюционного моделирования.

Научная новизна полученных результатов состоит в следующем:

- продемонстрирована возможность синтеза систем управления при помощи комбинации генетического алгоритма и искусственных нейронных сетей;
- разработаны методики выбора структуры нейросетевых компонентов систем управления;
- решена задача синтеза нейросетевых систем управления с заданными частотными характеристиками;
- выполнено сопоставление преимуществ и трудностей реализации нейроконтроллеров на базе кибернетического подхода и в рамках идеологии ПИД-регуляторов.

Практическая ценность диссертационной работы состоит в разработке универсальной процедуры синтеза нейросетевых компонентов систем управления, которая была успешно применена при построении системы управления электроприводом с нагрузкой типа «пара трения», моделей потребления электрической энергии промышленным предприятием. По мере компьютеризации учета потребления тепловой энергии,

газа, холодной и горячей воды с помощью разработанной процедуры можно будет синтезировать прогностические модели потребления.

Результаты научных исследований, приведенные в диссертации, были апробированы и внедрены в КПФ «ХАКИС» и ЧФ «Адаптивные системы», а также использованы при подготовке и выполнении в Харьковском государственном политехническом университете научно-исследовательской работы М3407 «Разработка принципов и алгоритмов управления энергетическими и электромеханическими системами с использованием искусственных нейронных сетей и генетических алгоритмов», осуществляемой в рамках координационного плана Минобразования Украины «Научные основы усовершенствования производства, передачи и использования электроэнергии».

Достоверность разработанной процедуры синтеза нейросетевых компонентов систем управления подтверждается многочисленными примерами успешного решения задачи построения нейроэмуляторов и нейроконтроллеров различных динамических объектов. Сравнительная эффективность предложенных алгоритмов доказана путем сопоставления результатов решения ими типовых задач с результатами, достигнутыми с помощью других, ранее известных алгоритмов.

Основные результаты диссертации опубликованы в соавторстве с Сергеевым С. А., Клепиковым В. Б. и Вороновским Г. К. В коллективной монографии [52] автору принадлежит методика и результаты синтеза нейросетевых систем управления, а также методика защиты интерполяционной матрицы при обучении радиально-базисных нейронных сетей. В докладе [81] автору принадлежат методика и результаты обучения ПИД-нейроконтроллера на базе прямонаправленной искусственной нейронной сети типа многослойный перцептрон. В статье [79] автору принадлежит структура ПИД-нейроконтроллера, а также выводы о применимости методики синтеза нейросетевых систем управления на базе

генетического алгоритма для решения задач управления нелинейными объектами.

Результаты исследований, приведенные в диссертации, были обнародованы на 3-х республиканских научных конференциях («Проблемы автоматизированного электропривода», 1994—97 гг., «microCAD. Информационные технологии: наука, техника, технология, образование, здоровье», 1995—97 гг., «Автоматика», 1996—97 гг.), а также 3-х международных конференциях во Франции, 1995 г., Германии и Польше, 1996 г.

Результаты диссертации опубликованы в одной коллективной монографии, 3-х статьях в научных журналах, а также в материалах 6-ти конференций.

ГЛАВА 1

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ — НОВАЯ ПАРАДИГМА В УПРАВЛЕНИИ

Сегодня, как и сто лет назад, несомненно, что мозг работает более эффективно и принципиально другим образом, чем любая вычислительная машина, созданная человеком. Именно этот факт в течение стольких лет побуждает и направляет работы ученых всего мира по созданию и исследованию искусственных нейронных сетей (ИНС).

К первым попыткам раскрыть секрет высокой эффективности мозга можно отнести работу Рамон-и-Кахаля (1911) [1], в которой была высказана идея о нейроне как структурной единице мозга. Однако нейрон имеет на 5—6 порядков меньшую скорость срабатывания, чем полупроводниковый логический элемент. Как показали более поздние исследования, секрет высокой производительности мозга заключается в огромном количестве нейронов и массивных взаимосвязях между ними.

Сеть нейронов, образующая человеческий мозг, представляет собой высокоэффективную, комплексную, нелинейную, существенно параллельную систему обработки информации [2]. Она способна организовать свои нейроны таким образом, чтобы реализовать восприятие образа, его распознавание или управление движением, во много раз быстрее, чем эти задачи будут решены самыми современными компьютерами [3].

ИНС является упрощенной моделью мозга. Она строится на основе искусственных нейронов, которые обладают тем же основным свойством, что и живые: пластичностью. Использование структуры мозга и пластичности нейронов делает ИНС универсальной системой обработки информации.

В общем случае ИНС — это машина, моделирующая способ работы мозга. Обычно ИНС реализуются в виде электронных устройств

или компьютерных программ. Среди многих можно выделить определение ИНС как адаптивной машины, данное в [4]:

Искусственная нейронная сеть — это существенно параллельно распределенный процессор, который обладает естественной склонностью к сохранению опытного знания и возможностью предоставления его нам. Она сходна с мозгом в двух аспектах:

- 1. Знание приобретается сетью в процессе обучения;*
- 2. Для сохранения знания используются силы межнейронных соединений, называемые также синаптическими весами.*

Процедура, используемая для осуществления процесса обучения, называется алгоритмом обучения. Ее функция состоит в модификации синаптических весов ИНС определенным образом так, чтобы она приобрела необходимые свойства.

Модификация весов является традиционным способом обучения ИНС. Такой подход близок к теории адаптивных линейных фильтров, которые уже давно и успешно применяются в управлении. Однако для ИНС существует еще и возможность модификации собственной топологии, основывающаяся на том факте, что в живом мозге нейроны могут появляться, умирать и менять свои связи с другими нейронами.

Из сказанного выше становится ясно, что ИНС реализуют свою вычислительную мощь, благодаря двум основным своим свойствам: существенно параллельно распределенной структуре и способности обучаться и обобщать полученные знания. Под свойством обобщения понимается способность ИНС генерировать правильные выходы для входных сигналов, которые не были учтены в процессе обучения (тренировки). Эти два свойства делают ИНС системой переработки информации, которая решает сложные многомерные задачи, непосильные другим техникам.

1.1. Этапы развития искусственных нейронных сетей

Если опустить чисто биологический этап, то можно сказать, что современная эпоха развития ИНС началась в 1943 году с пионерской работы Мак-Каллока и Питтса [5]. В этой работе Мак-Каллок и Питтс ввели понятие «порогового логического нейрона» и описали логическую модель ИНС.

В 1948 году была опубликована известная книга Винера «Кибернетика» [6], описывающая ряд важных концепций управления. В более поздних изданиях в нее были добавлены разделы, посвященные обучению, самоорганизации и нейрокибернетике.

Следующим событием, продвинувшим развитие ИНС, стал выход в свет книги Хебба [7], в которой впервые было представлено физиологическое обучающее правило модификации синапсов. Хебб предположил, что соединения в мозге непрерывно меняются по мере того, как организм обучается новым функциональным задачам, и что таким образом создаются нейронные ансамбли. Хебб следовал сделанному ранее предположению Рамон-и-Кахалы и ввел «постулат обучения», который гласит:

Эффективность (усиление) переменного синапса между двумя нейронами увеличивается в процессе повторяющейся активации одного нейрона другим через этот синапс.

Книга Хебба стала настоящим источником вдохновения для создания обучающихся и адаптивных систем. Работы Рочестера, Холланда, Хейбта и Дуде в 1956 г. [8] были, наверное, первыми попытками использовать компьютерное моделирование для тестирования теории ИНС, базирующейся не постулате обучения Хебба. Полученные ими результаты показали, что не только активация, но и торможение должны присутствовать в процессе обучения. В тот же год Уитли [9] проде-

монстрировал, что ИНС с модифицируемыми синапсами может быть обучена классифицировать простые наборы бинарных шаблонов по соответствующим классам. В более поздней своей работе Уитли [10] также выдвинул гипотезу о том, что эффективность переменного синапса в нервной системе зависит от статистических взаимоотношений между изменяющимися состояниями с обеих сторон от синапса, тем самым проведя связь с информационной теорией Шеннона.

В 1952 году вышла в свет книга Эшби «Design for a Brain: The Origin of Adaptive Behavior» [11]. Одна из главных идей этой книги состоит в том, что адаптивное поведение живых систем не является наследственным, а скорее появляется в процессе обучения, и что обычно в процессе обучения поведение живых систем улучшается.

Важным моментом в истории ИНС было написание Минским в 1954 году докторской диссертации, посвященной ИНС [21]. В 1961 году он опубликовал статью «Steps Toward Artificial Intelligence» [13], которая была посвящена искусственному интеллекту и содержала большой раздел, посвященный тому, что сейчас называют ИНС.

Необходимо отметить огромную важность работ фон Неймана по созданию цифровых компьютеров для развития всего кибернетического направления. Особое значение для теории ИНС имела его идея избыточности, которая подтолкнула Винограда и Коуэна (1963) [14] к созданию распределенного избыточного представления ИНС.

Примерно 15 лет спустя появления эпохальной статьи МакКаллока и Питтса, в 1958 году Розенблатт в своей работе, посвященной перцептрон [15], предложил новый подход к решению задачи распознавания шаблонов. В ней он ввел так называемую теорему о сходимости перцептрона, доказательство которой появилось лишь в 1960 году. В 1960 Видроу и Хофф предложили метод наименьших квадратов и ис-

пользовали его для описания Adaline [16]. Основное различие между перцептроном и Adaline лежало в процедурах их обучения.

Одной из первых обучаемых ИНС со слоистой структурой был Madaline, предложенный Видроу и его студентами в 1962 г. [17]. Не менее важным событием было введение в 1967 Коуэн сигмоидальной активационной функции для модели логического нейрона [18].

В течение классического периода перцептрона 60-х годов казалось, что ИНС могут делать все. Однако в 1969 вышла книга Минского и Пейперта [19], в которой строго математически было доказано существование фундаментальных ограничений возможностей однослойного перцептрона. Также было показано, что однослойная сеть любой размерности не может решать задачи доступные многослойным конфигурациям.

Для многослойных перцептронов десятилетие 70-х не принесло ничего нового. Это было связано отчасти с низким уровнем финансирования работ, отчасти с отсутствием персональных компьютеров. Негативное влияние оказало также не подтвердившееся позже мнение об аналогии между спиновыми стеклами и ИНС. С другой стороны, в это же время были получены значительные результаты в развитии самоорганизующихся карт на базе соревновательного обучения. В 1976 году фон Мальсбург и Вильшоу опубликовали первую работу [20] по самоорганизующимся картам, продемонстрировав действующую самоорганизующуюся ИНС.

С начала 80-х наступил новый период в развитии ИНС. Основными событиями были разработка Хопфилдом ИНС с полностью связанной структурой и оригинальным алгоритмом настройки весов. В 1982 в продолжение исследований фон Мальсбурга выходит работа Кохонена [21], посвященная самоорганизующимся картам. Но настоящий прорыв в применении ИНС для решения практических задач (в том чис-

ле и управления) был сделан после того, как в 1986 году Румельхарт, Хинтон и Вильямс [22] описали алгоритм обратного распространения. Это был первый эффективный алгоритм обучения многослойных перцептронов любой структуры (и, как оказалось позже, не только их).

Интересно, что в еще 1985 г. независимо вышли две работы Паркера [23] и Ле-Кана [24] с описанием алгоритма обратного распространения. Однако оказалось, что впервые этот алгоритм был детально описан в 1974 в докторской диссертации Вербоса [25]. К сожалению, тогда на него никто не обратил внимания, и потребовалось более десяти лет, чтобы эта простая, но изящная схема была переоткрыта.

В 1988 г. произошло последнее на сегодняшний день крупное открытие в теории ИНС, связанное с введением Брумхедом и Лоуе [26] РБФ-сетей. Это — альтернативная многослойному перцептрону схема слойной прямонаправленной сети, использующая скрытые нейроны с радиально-базисной активационной функцией. Идеологически идея радиально-базисных функции связана с методом потенциальных функций, предложенным в 1964 г. Башкировым, Браверманом и Мучником [27]. В своих работах Брумхед и Лоуе не только предложили новый метод синтеза ИНС, но и уделили огромное внимание установлению связей между ИНС и классическими методами численного анализа, а также теорией линейных адаптивных фильтров.

1.2. Модель нейрона

Искусственный нейрон (или просто нейрон) является элементарным функциональным модулем, из множества которых строятся ИНС. Он представляет собой модель живого нейрона, однако лишь в смысле осуществляемых им преобразований, а не способа функционирования. Существуют логические, непрерывные и импульсные [28] модели ней-

рона. Логические модели нейрона (в частности, описываемый картой Вена формальный нейрон) активно исследовались в 60—70-х годах, но не получили дальнейшего развития. Импульсные модели более близки к физической природе процессов, происходящих в нервной клетке, однако их теория не так развита как у непрерывных, и они все еще не находят широкого применения.

Непрерывная модель нейрона работает следующим образом. Входные сигналы поступают на блоки, реализующие функцию синапсов. Каждый из них характеризуется своим весовым коэффициентом (синаптическим весом). Положительные значения весов w_{kj} соответствуют возбуждательным синапсам, отрицательные — тормозным. Взвешенные входные сигналы подаются на линейный сумматор, после чего результат их сложения поступает на блок активационной функции. Обычно активационная функция ограничивает выходной сигнал нейрона в диапазоне $[0,1]$ или $[-1,1]$. Модель нейрона также включает в себя сдвиг b , который добавляется к входному сигналу блока активационной функции.

Математически модель нейрона описывается следующими зависимостями:

$$v_k = \sum_{j=1}^l w_{jk} z_j ; \quad (1.1)$$

$$s_k = v_k + b_k ; \quad (1.2)$$

$$y_k = \varphi(s_k) ; \quad (1.3)$$

где z_1, z_2, \dots, z_l — входные сигналы нейрона,

$w_{1k}, w_{2k}, \dots, w_{lk}$ — синаптические веса k -го нейрона,

v_k — выход линейного сумматора,

b_k — сдвиг,

$\varphi(\cdot)$ — активационная функция,

y_k — выходной сигнал нейрона.

Эту запись можно упростить, введя дополнительный входной сигнал $z_0=1$ и вес $w_{0k}=b_k$. Тогда (2) примет вид:

$$s_k = \sum_{j=0}^l w_{jk} x_j . \quad (1.4)$$

В дальнейшем w_{0k} условно будет называться сдвигом. На рис. 1.1 изображена функциональная схема такой модели искусственного нейрона непрерывного типа.

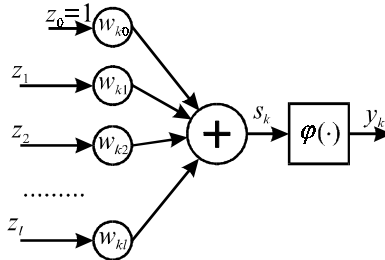


Рис. 1.1
Модель нейрона

Активационная функция нейрона $\varphi(\cdot)$ определяет нелинейное преобразование, осуществляемое нейроном. Существует множество видов активационных функций, но более всего распространены следующие четыре:

1. Пороговая функция. На рис. 1.2,а приведен ее график.

$$\varphi(s) = \begin{cases} 1, & \text{если } s \geq 0 \\ 0, & \text{если } s < 0 \end{cases} . \quad (1.5)$$

Это — первая из введенных активационных функций, она была описана в работе Мак-Каллока и Питтса [5]. В честь этого модель нейрона с пороговой активационной функцией называется моделью Мак-Каллока-Питтса. Также в литературе обычно ссылаются на непрерыв-

ную модель нейрона типа $\varphi(s)$, где s определяется по (1.4), как на модель Мак-Каллока-Питтса.

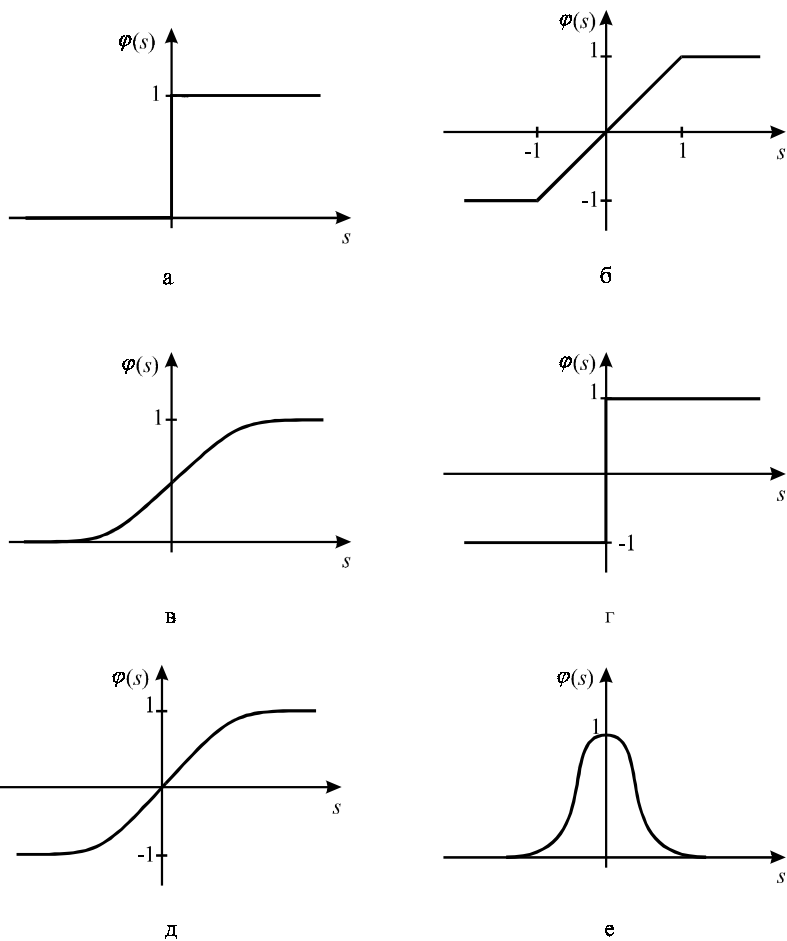


Рис. 1.2

Типы активационных функций.

а), г) — пороговая; б) — линейная; в) — сигмоидальная;
 д) — тангенциальная; е) — радиально-базисные активационные функции

2. Кусочно-линейная функция. Она изображена на рис. 1.2,б и описывается следующей зависимостью:

$$\varphi(s) = \begin{cases} 1, & \text{если } s \geq a \\ s, & \text{если } a > s > -a \\ -1, & \text{если } a \leq -a \end{cases} \quad (1.6)$$

В данном случае $a=1$, и коэффициент наклона линейного участка выбран единичным, а вся функция может интерпретироваться как аппроксимация нелинейного усилителя. При бесконечно большом коэффициенте наклона линейного участка функция вырождается в пороговую.

В большинстве типов ИНС используются нейроны с линейной активационной функцией $\varphi(s) = s$, представляющей собой частный случай (1.6) с неограниченным линейным участком.

3. Сигмоидальная функция. Это наиболее широко используемый тип активационной функции. Она была введена по аналогии с пороговой функцией, но везде является строго монотонно возрастающей, непрерывной и дифференцируемой (см. рис. 1.2,в). Дифференцируемость является важным свойством для анализа ИНС и некоторых методов их обучения.

В общем виде сигмоидальная активационная функция описывается зависимостью:

$$\varphi(s) = \frac{1}{1 + e^{-as}}, \quad (1.7)$$

где a — параметр, определяющий наклон функции.

Варьированием его могут быть получены разные виды сигмоида. Наиболее часто используется $a=1$. В случае бесконечно большого a сигмоидальная функция вырождается в пороговую.

Помимо перечисленных функций, изменяющихся в диапазоне $[0,1]$, вводятся также их аналоги с областью значений $[-1,1]$. Так например (см. рис. 1.2,г), пороговая функция может быть переопределена как

$$\varphi(s) = \begin{cases} 1, & \text{если } s > 0 \\ s, & \text{если } s = 0 \\ -1, & \text{если } s < 0 \end{cases}. \quad (1.8)$$

То есть

$$\varphi(s) = \text{sign}(s). \quad (1.9)$$

Вместо сигмоидальной активационной функции широко применяется гиперболический тангенс, обладающий аналогичными свойствами (см. рис. 1.2,д)

$$\varphi(s) = \text{th}(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}. \quad (1.10)$$

Нечетность этой функции делает ее удобной для решения задач управления.

4. Во введенных Брумхедом и Лоуе [26] РБФ-сетях в качестве активационной применяется функция Гаусса (см. рис. 1.2,е)

$$\varphi(s) = e^{-\frac{s^2}{\sigma^2}}. \quad (1.11)$$

Однако в отличие от (1.4) ее аргумент рассчитывается по формуле:

$$s = \|\mathbf{z} - \mathbf{c}\|, \quad (1.12)$$

где \mathbf{z} — вектор входных сигналов нейрона,

\mathbf{c} — вектор координат центра окна активационной функции,

σ — ширина окна,

норма $\|\cdot\|$ — евклидово расстояние.

В теории ИНС активационные функции типа

$$\varphi(\mathbf{z}) = \varphi(\|\mathbf{z} - \mathbf{c}\|) \quad (1.13)$$

называются радиально-базисными функциями (РБФ), а основанные на них сети — РБФ-сетями (RBF — radial basis function).

1.3. Архитектура искусственных нейронных сетей

В общем случае понятие «искусственная нейронная сеть» охватывает ансамбли нейронов любой структуры, однако практическое применение нашли только некоторые из них. Это объясняется тем, что архитектура ИНС непосредственно связана с методом ее обучения. Даже различные этапы развития ИНС определялись появлением новых архитектур сетей и специально разработанных для них методов обучения.

На сегодняшний день можно выделить четыре основные разновидности архитектуры ИНС.

1. Однослойные прямонаправленные сети. Слоистой называется ИНС, состоящая из групп нейронов, разделенных по слоям. ИНС, содержащая k слоев, называется k -слоистой. Если сигналы в сети распространяются только по направлению из начала в конец, то такая ИНС называется прямонаправленной.

На рис. 1.3 изображена однослойная прямонаправленная ИНС.

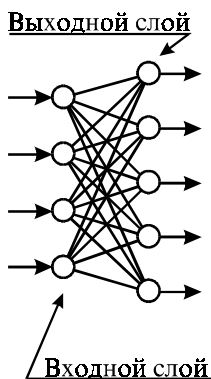


Рис. 1.3

Прямонаправленная сеть с одним слоем нейронов

Она включает в себя слой входных нейронов и слой выходных. Нейроны входного слоя просто ретранслируют сигналы на выходной слой, не преобразуя их. В выходном слое происходит преобразование сигналов и формирование реакции сети. Необходимо отметить, что нейроны ИНС также иногда называются узлами или вычислительными модулями. Количество нейронов в одном слое определяет размер слоя.

В отличие от принятой методики обозначения, такие ИНС называются однослойными, а не двухслойными. Этим подчеркивается, что вычисления производятся лишь одним слоем сети.

2. Многослойные прямонаправленные сети. Они характеризуются наличием одного или нескольких скрытых слоев, осуществляющих преобразование информации. Нейроны скрытого слоя называются скрытыми нейронами или скрытыми узлами. Использование скрытых слоев позволяет ИНС осуществлять нелинейные преобразования вход-выход любой сложности или извлекать из входных данных статистические показатели высоких порядков. Эти уникальные свойства многослойных сетей особенно проявляются при высокой размерности пространства входных сигналов.

На рис. 1.4 представлена схема трехслойной прямонаправленной ИНС с одним скрытым слоем. Для описания такой сети используется запись $NN3-5-2$. Здесь 3 — размер входного слоя сети, 5 — скрытого, и 2 — выходного. В общем случае прямонаправленная ИНС с g входными нейронами, q выходными нейронами и n скрытыми слоями размера h_i обозначается $NNg-h_1-h_2-\dots-h_n-q$.

Нейроны входного слоя в таких сетях просто ретранслируют входные сигналы на первый скрытый слой, не преобразуя их. В скрытых нейронах последовательно, слой за слоем, происходит нелинейное преобразование сигналов. Сигналы с последнего скрытого слоя поступают на нейроны выходного слоя, которые формируют реакцию сети.

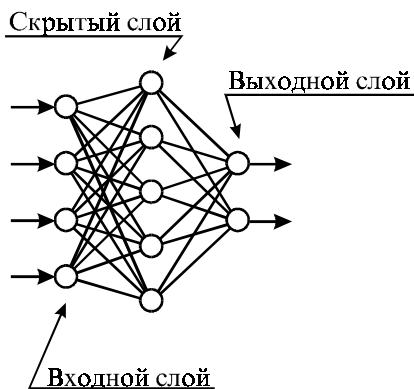


Рис. 1.4

Полностью связанная трехслойная прямонаправленная сеть

Формально, не существует ограничений на типы активационных функций нейронов различных слоев ИНС или даже одного слоя, однако обычно все скрытые нейроны выбираются одного типа. Выходной слой ИНС может состоять из нейронов с тем же типом активационной функции, что и у нейронов скрытого слоя, но наиболее распространенным является модель прямонаправленной сети с линейными выходными нейронами. ИНС этого типа с активационными функциями нейронов скрытого слоя (1.7, 1.10) называются многослойными перцептронами (МСП). МСП нашли широкое применение при решении различных задач и являются одним из главных объектов теоретических исследований.

Изображенная на рис. 1.4 ИНС называется полностью связанной прямонаправленной ИНС, так как каждый нейрон из одного слоя связан со всеми нейронами следующего слоя. Широкое применение также находят частично связанные прямонаправленные ИНС, в которых нейроны одного слоя связаны только с определенной частью нейронов следующего слоя. Такая архитектура позволяет заложить в ИНС априорные знания о желаемом законе обработки сигналов в сети.

Трехслойные прямонаправленные ИНС широко используются для решения задач классификации, распознавания образов, аппроксимации и управления.

3. Рекуррентные сети. Этот тип ИНС отличается существованием обратных связей и элементов временной задержки сигнала.

Наиболее простым случаем рекуррентной сети является один слой нейронов, охваченный обратными связями. При этом каждый нейрон получает задержанные выходные сигналы всех остальных нейронов.

На рис. 1.5 представлена рекуррентная ИНС, содержащая скрытый слой нейронов. В этом случае каждый нейрон получает, кроме входных сигналов, еще и все выходные сигналы сети. Часть ИНС, охваченная обратными связями может иметь и большее количество скрытых слоев.

Наличие обратных связей и элементов временной задержки сигналов придает рекуррентным сетям собственные нелинейные динамические свойства. Это также сказывается на их способности к обучению. Тренировка рекуррентных сетей требует учета их динамических свойств.

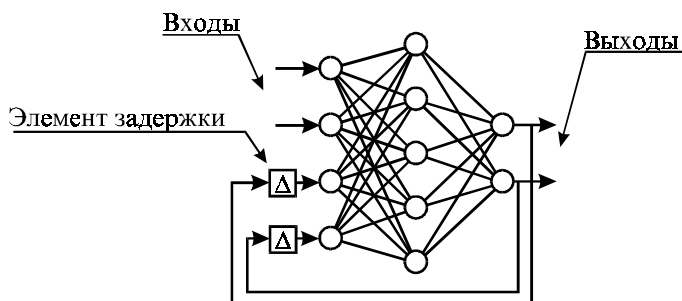


Рис. 1.5

Рекуррентная сеть с одним скрытым слоем

Одним из главных применений рекуррентных ИНС являются нейроэмуляторы динамических объектов, то есть их нейросетевые модели. Такие сети могут также использоваться для решения задач аппроксимации временных последовательностей, классификации, распознавания образов и управления.

4. Полностью связанные сети. Характерным признаком ИНС этого типа является наличие связей между всеми нейронами.

Наиболее известной разновидностью полностью связанных сетей являются сети Хопфилда (см. рис. 1.6). В них каждый нейрон имеет двусторонние связи со всеми остальными нейронами сети. В общем случае сеть Хопфилда имеет симметричную кольцевую структуру, в ней нельзя выделить скрытые нейроны и единое направление распространения сигналов. Работа полностью связанной ИНС и обмен данными контролируется одним главным нейроном.

Сеть Хопфилда является примером полностью связанной динамической сети, основывающейся на принципах самоорганизации, однако в ней не используются в явном виде элементы временной задержки.

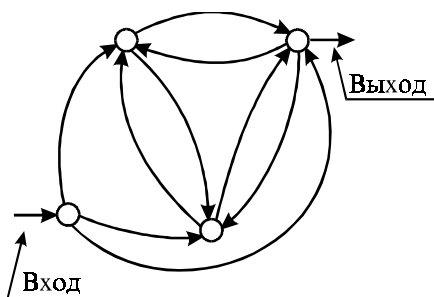


Рис. 1.6

Полностью связанная сеть Хопфилда

Другим примером служат решетчатые сети (см. рис. 1.7). Они представляют собой массив нейронов, каждый из которых связан с входными нейронами. Размерность массива нейронов определяет раз-

мерность решетчатой сети. Такая ИНС является прямонаправленной, так как в ней нет обратных связей, однако в ней нельзя выделить скрытые элементы или слои.

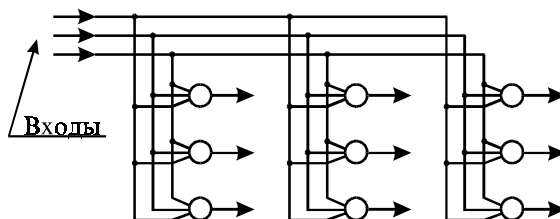


Рис. 1.7

Одномерная решетчатая сеть из трех нейронов

Полностью связанные сети находят особо широкое применение при решении задач классификации и распознавания образов.

1.4. Обучение искусственных нейронных сетей

Как и их биологические прообразы, ИНС могут обучаться, то есть улучшать свою работу под воздействием окружающей среды, изменяющей ее параметры. Существует множество определений термина «обучение», однако применительно к ИНС наиболее подходит следующее, данное Менделем и Маклареном в [29]:

Обучение — это процесс, при котором свободные параметры нейронной сети адаптируются в результате ее непрерывной стимуляции внешним окружением. Тип обучения определяется тем способом, которым производятся изменения параметров.

В современной литературе кроме термина «обучение» также используются равноправные понятия «тренировка сети» и «настройка параметров сети».

В общем можно выделить два основных вида обучения [30]: контролируемое обучение (supervised learning) и самообучение (self-

organized learning). Первый вид подразумевает наличие «учителя», который наблюдает реакцию сети и направляет изменения ее параметров. Во втором случае сеть самоорганизуется под действием внешней среды и изучает ее самостоятельно, без помощи «учителя». Самообучение свойственно задачам распознавания образов и классификации. При решении задач управления обычно используется контролируемое обучение ИНС.

Существует две разновидности контролируемого обучения: прямое контролируемое обучение и стимулированное обучение (reinforcement learning). Так как первый вид появился раньше второго и более распространен, то обычно на него ссылаются просто как на контролируемое обучение.

На рис. 1.8 представлена схема прямого контролируемого обучения. Присутствующий на ней «учитель» обладает знаниями о внешней среде, представленными в виде набора образцов вход-выход. Пара таких образцов называется тренировочным шаблоном, который включает в себя, соответственно, входной шаблон и выходной шаблон. Множество этих пар данных называется тренировочным набором шаблонов.

ИНС изначально никакими знаниями не обладает. В процессе обучения «учитель» и сеть подвергаются воздействию со стороны внешней среды, то есть на их входы поступает тренировочный сигнал, совпадающий с одним из входных шаблонов. «Учитель» сообщает сети, какая должна быть правильная (желаемая) реакция на поступившее воздействие, выдавая соответствующий выходной шаблон. На основании величины ошибки между реальным и желаемым выходами сети по определенному правилу проводится настройка ее синаптических связей. Повторяя этот процесс итеративно можно настроить ИНС так, что она будет эмулировать «учителя», то есть его знания о внешней среде перейдут к ней.

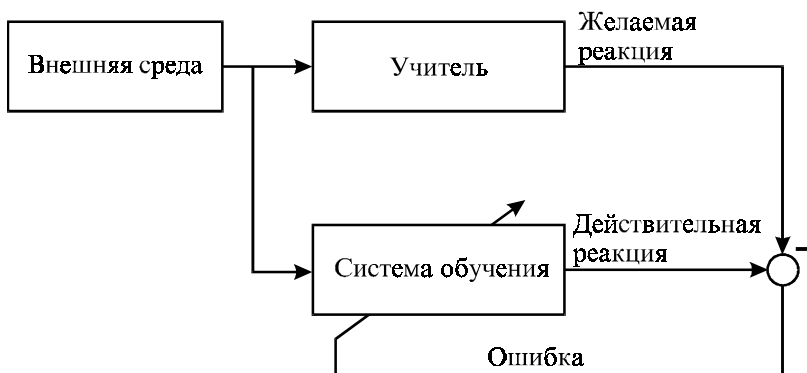


Рис. 1.8

Схема прямого контролируемого обучения

Типичными задачами, решаемыми с помощью прямого обучения, являются аппроксимация неизвестной функции, описываемой набором данных, и идентификация динамического объекта. В этих задачах известны входные сигналы и правильная реакция на них, то есть тренировочный набор шаблонов.

Наиболее известным методом прямого контролируемого обучения слонных прямонаправленных ИНС является алгоритм обратного распространения ошибки (backpropagation algorithm) [30], представляющий собой обобщение метода наименьших квадратов. Однако, по такой схеме могут работать и другие методы.

Стимулируемое обучение не использует знания «учителя» о желаемом выходе ИНС, вместо этого обучение проводится по результату оценки проводимого сетью преобразования вход-выход. Оценку выполняет внешнее окружение после подачи на вход сети тренировочного воздействия (см. рис. 1.9). При этом настройка параметров ИНС проводится так, чтобы максимизировать скалярный индекс этой оценки, называемый стимулом (reinforcement signal).

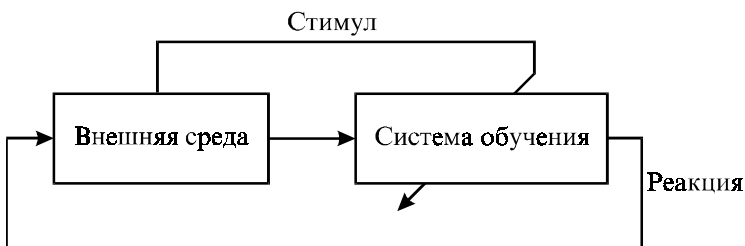


Рис. 1.9

Схема стимулируемого обучения

Идея такого способа опирается на реальный процесс обучения, проходящий у живых существ. В психологии он известен, как *закон действия Торндайка*. Применительно к стимулируемому обучению ИНС этот закон может быть перефразирован следующим образом [31]:

Если действие, предпринятое системой обучения, приводит к удовлетворительному результату, то тенденция системы проводить это же действие увеличивается (система стимулируется). В противном случае тенденция производить такое действие уменьшается.

Наиболее характерным примером системы стимулируемого обучения является адаптивная система управления. В ней обучаемой частью является контроллер, а объект управления, внешние воздействия и сигналы задания выступают его внешней средой. В результате воздействия из этой среды контроллер вырабатывает определенный управляющий сигнал, который переводит объект управления в новое состояние. При этом качество управления можно оценить лишь по выходному сигналу объекта. Так как требуемая реакция контроллера, обеспечивающая заданное состояние объекта, заранее неизвестна, то нельзя сформировать тренировочный набор шаблонов, и, следовательно, применить прямое контролируемое обучение. В этом случае возможно лишь стимулируемое обучение контроллера по качеству работы всей

системы управления в целом, то есть по оценке состояния внешней среды.

Возможны две реализации стимулируемого обучения: с непосредственным стимулированием и с задержанным стимулированием. В первом случае оценка работы и соответствующая ей настройка параметров ИНС проводится на каждом шаге работы системы обучения. Во втором случае, при задержанном стимулировании настройка сети выполняется так, чтобы максимизировать кумулятивную оценку работы системы на определенной последовательности шагов.

Хотя непосредственное стимулирование более свойственно классическим схемам адаптивного управления, в последнее время большое внимание уделяется методам задержанного стимулирования. Для сходимости метода непосредственного стимулирования необходимо, чтобы каждое последующее положение внешней среды определялось лишь ее предыдущим положением и предыдущим воздействием, оказанным ИНС [30]. При задержанном обучении такое ограничение снимается. С другой стороны, гораздо проще сформулировать критерий определяющий оптимальное поведение системы управления в целом, по результатам какого-то действия, чем критерий, направляющий ее движение на каждом шагу. В общем, можно заметить, что система, обучающаяся лишь по результатам своего взаимодействия с внешней средой, является более интеллектуальной, чем использующая дополнительную информацию «учителя».

Существует множество модификаций классического алгоритма обратного распространения ошибки для использования его как метода прямого или стимулируемого обучения ИНС. Однако существенной проблемой, возникающей при использовании его и других методов настройки параметров ИНС, основанных на методе наименьших квадратов или наискорейшего спуска, является их локальность. В то же время

целевая функция (суммарная ошибка на тренировочном наборе шаблонов или кумулятивная оценка эффективности работы обучаемой системы) не унимодальная. Количество локальных оптимумов для большинства практических задач обучения исчисляется миллионами при размерности поискового пространства порядка 10^2 — 10^3 . Вследствие этого результат обучения зависит от правильности выбора стартовой точки, и возникает необходимость многократного повторения процедуры настройки параметров ИНС.

Перечисленные проблемы могут быть решены при использовании методов глобальной оптимизации. Наиболее эффективным из них является генетический алгоритм (ГА) [32]. Рассматривая ИНС как единый набор параметров, ГА способен осуществлять ее оптимальную настройку при размерности поискового пространства достаточной для решения большинства практических задач. При этом спектр рассматриваемых приложений гораздо превосходит возможности алгоритма обратного распространения ошибки [33].

В последние десять лет разработано множество способов контролируемого обучения ИНС с помощью ГА. Полученные результаты доказывают большие возможности такого симбиоза. Совместное использование ИНС и ГА алгоритмов имеет и идеологическое преимущество потому, что они относятся к методам эволюционного моделирования и развиваются в рамках одной парадигмы заимствования техникой природных методов и механизмов как наиболее оптимальных.

1.5. Свойства искусственных нейронных сетей

1.5.1. Свойство обобщения.

В процессе контролируемого обучения синаптические веса ИНС настраиваются так, чтобы она аппроксимировала выходные данные

хранящиеся в предъявляемых ей тренировочных шаблонах. В конце процесса обучения сеть запоминает все тренировочные данные, то есть правильно реагирует на тестовые входы. Однако кроме способности помнить, сеть обладает свойством обобщать тренировочные данные.

Используемый здесь термин обобщение заимствован из психологии, и для обученной ИНС обозначает явление, когда осуществляемое ею преобразование вход-выход верно как для тренировочных данных, так и для тестовых данных, не использовавшихся при тренировке. Обобщение возникает в ИНС как следствие хорошей нелинейной интерполяции входных данных, главным образом благодаря непрерывности и гладкости активационной функции самих нейронов.

На рис. 1.10 проиллюстрировано свойство обобщения для гипотетического трехслойного перцептрона с одним входом и одним выходом. Кривая, изображенная на рисунке, отображает нелинейное преобразование осуществляемое ИНС. Точки, обозначенные как «тренировочные данные», использовались при обучении ИНС, остальные точки, помеченные «обобщение», получены как результат интерполяции, проводимой сетью.

ИНС не всегда может обобщать данные. На рис. 1.11 изображено преобразование, осуществляемое ИНС, обученной на том же тренировочном наборе. Видно, что в этом случае сеть не может корректно интерполировать входные данные между тренировочными. Это объясняется тем, что ИНС была перетренирована, то есть она запомнила тренировочные данные как таблицу и выдает корректные данные лишь при точном указании «адреса» в ней.

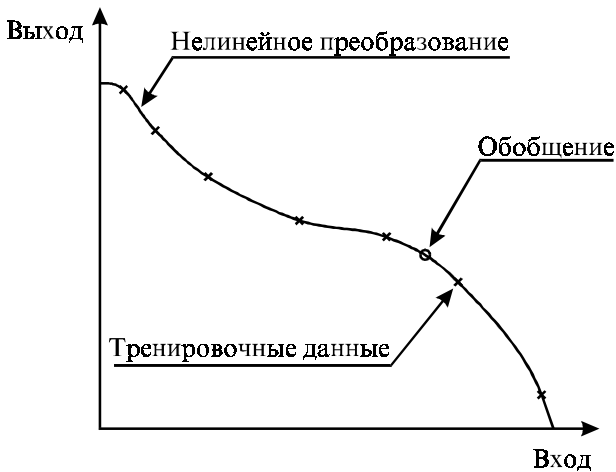


Рис. 1.10

Явление обобщения в нейронной сети

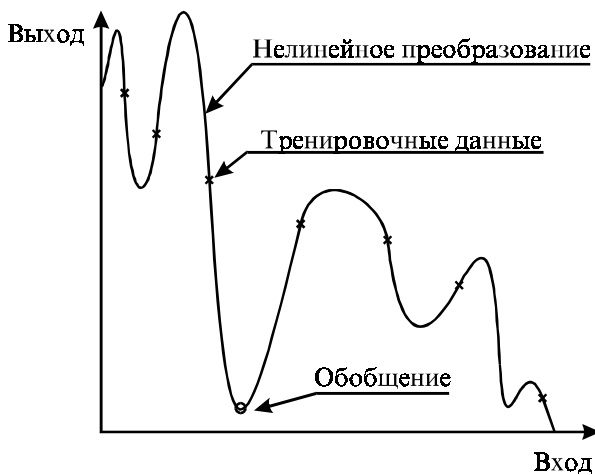


Рис. 1.11

Низкий уровень обобщения в перетренированной нейронной сети

При низком уровне обобщения преобразование, осуществляемое ИНС, недостаточно гладко. Как указано в [34], гладкость преобразова-

ния в ИНС непосредственно связано с критерием выбора модели, известным как «Бритва Оккама», то есть с необходимостью выбрать простейшую модель в условиях отсутствия априорной информации. В данном случае под простейшей подразумевается наиболее гладкая функция, наилучшим образом аппроксимирующая рассматриваемое преобразование. Требование гладкости обеспечивает хорошие интерполяционные свойства ИНС и гарантирует ей минимальную сложность структуры. Последний показатель прямо влияет на объем вычислений, выполняемых при обучении.

Можно выделить два необходимых, но не достаточных условия существования обобщения в ИНС. Первое — это гладкость того нелинейного преобразования, которое сеть должна запомнить. То есть небольшим приращениям входных сигналов должны соответствовать небольшие приращения выходных. Это условие не всегда выполняется, особенно если тренировочные данные получены в ходе реальных экспериментов и содержат большое количество помех.

Вторым условием является достаточная репрезентативность тренировочного набора данных. Обычно задача, решаемая с помощью ИНС, содержит объем данных намного превышающий вычислительные возможности компьютеров, поэтому для ее обучения используется только некоторая тренировочная выборка. Отбор данных в тренировочный набор необходимо проводить так, чтобы, с одной стороны, его можно было обрабатывать за приемлемое время, а с другой стороны, он был достаточно большим и полно описывал исходную задачу.

Подходя упрощенно, можно классифицировать проводимое ИНС обобщение, как интерполяцию или экстраполяцию в зависимости от того, как во входном пространстве расположены тестовые данные относительно запомненных ею тренировочных шаблонов. Учитывая, что интерполяция, выполняется проще и точнее, чем экстраполяция, можно

утверждать, что ИНС тем лучше обобщает, чем плотнее и равномернее расположены тренировочные данные во входном пространстве. Если тестовые данные всегда будут оказываться между близко расположенными тренировочными шаблонами, то сеть сможет проводить корректное обобщения, интерполируя, а не экстраполируя, входные данные.

При выполнении прочих условий огромное влияние на качество обобщения, проводимого ИНС, оказывает сам процесс обучения. Первая из существующих угроз потери обобщения — это возможность недотренировать ИНС, прекратив обучение сразу после достижения приемлемой ошибки на тренировочном наборе. В этом случае в сети могут остаться нейроны, не задействованные в запоминании тренировочных данных, и их вклад в преобразование, осуществляемое сетью, может мешать правильной интерполяции. Вторая угроза — риск перетренировать ИНС. Имеется в виду, что каждый нейрон перетренированной сети настроен так, чтобы максимально точно соответствовать всем шаблонам, предъявлявшимся в ходе тренировки. В результате этого действие активационных функций нейронов становится локальным, интерполяция по ним становится невозможна. Фактически, перетренированная ИНС превращается (как на рис. 1.11) в таблицу.

Выходом в данной ситуации является постоянный контроль по ходу процесса обучения за способностью ИНС обобщать. Для этого можно использовать расширенный набор тестовых данных, содержащий помимо тренировочных еще и близкие к ним дополнительные шаблоны.

1.5.2. Свойство универсальной аппроксимации.

В работах Кибенко [35], Фунахаши [36], и Хорника [37] было показано, что МСП может аппроксимировать непрерывную функцию любого типа. В [37] это свойство представлено в виде теоремы универсальной аппроксимации, которая гласит:

Пусть $\varphi(\cdot)$ — непрерывная непостоянная функция, и пусть C — множество непрерывных функций, заданных на \mathbb{R}^g . Тогда для любых $f \in C$ и $\varepsilon > 0$ существует такое целое h и набор вещественных α_i, b_i, w_{ij} , где $i=1, \dots, h$ и $j=1, \dots, g$, что

$$F(x_1, \dots, x_g) = \sum_{i=1}^h \alpha_i \varphi\left(\sum_{j=1}^g w_{ij} x_j + b_i\right) \quad (1.14)$$

является аппроксимацией функции $f(\cdot)$. То есть

$$\left\| F(x_1, \dots, x_g) - f(x_1, \dots, x_g) \right\| < \varepsilon \quad (1.15)$$

для всех $\{x_1, \dots, x_g\} \in \mathbb{R}^g$.

Это — обобщение известной теоремы об аппроксимации конечными рядами Фурье [30]. Важно, что активационной функцией $\varphi(\cdot)$ может быть любая непрерывная непостоянная функция, а не только перечисленные выше (1.5)—(1.11).

Теорема универсальной аппроксимации является теоремой существования, она определяет необходимые, но не достаточные условия аппроксимации произвольной непрерывной функции. Как видно, (1.14) описывает трехслойный перцептрон, однако существуют аналоги этой теоремы для случая многослойного перцептрона и других типов ИНС. Тем не менее, ни одна из них не утверждает оптимальность того или иного типа ИНС. Теорема универсальной аппроксимации не дает также рекомендаций по выбору размера скрытого слоя h . Поэтому вопрос о типе и размере ИНС должен решаться в каждом конкретном случае опытным путем.

В работе [38] была рассмотрена трехслойная прямонаправленная сеть с сигмоидальной активационной функцией нейронов скрытого слоя. Для нее было показано, что в случае решения задачи аппроксимации некоторой непрерывной функции $f(\cdot)$, общий риск r , определяемый

как среднеквадратичная интегральная ошибка между функцией и ее аппроксимацией, ограничен величиной

$$O\left(\frac{1}{h}\right) + O\left(\frac{hg}{N} \log N\right), \quad (1.16)$$

где h — размер скрытого слоя сети,

g — количество входных нейронов,

N — количество тренировочных шаблонов.

Риск r является оценкой возможности ИНС обобщать новые данные. Ограничение (1.16) показывает, что существуют два противоречивых условия, определяющие выбор размера скрытого слоя сети h :

- точность наилучшей аппроксимации (по данным из тренировочного набора), которая в соответствии с теоремой универсальной аппроксимации требует достаточно большого количества скрытых нейронов h ;
- точность эмпирической аппроксимации (по данным не вошедшим в тренировочный набор), которая требует, чтобы отношение размера скрытого слоя к размеру тренировочного набора h/N было достаточно мало.

Отсюда следует, что в случае МСП достижение определенной точности аппроксимации не требует экспоненциального увеличения размера тренировочного набора данных. Другим важным выводом является то, что для больших N ошибка оценивания ИНС ε_o имеет порядок hg/N , а значит для достаточного уровня обобщения размер тренировочного набора данных должен быть

$$N > \frac{hg}{\varepsilon_o}. \quad (1.17)$$

То есть необходимая величина N прямо пропорциональна общему количеству настроечных параметров сети.

В [30] показано также, что скорость сходимости общего риска r для ИНС выражается функцией размера тренировочного набора данных порядка $(1/N)^{1/2}$. В то же время в случае аппроксимации традиционными гладкими функциями, с ограниченными производными порядка $s > 0$, скорость сходимости имеет порядок $(1/N)^{2s/(2s+g)}$. Зависимость скорости от размерности входного пространства g представляет собой *проклятие размерности*, ограничивающее применимость традиционных гладких функций, в то же время ИНС не страдают этим недостатком. Это является еще одним важным свойством ИНС.

1.6. Применение искусственных нейронных сетей для решения задач управления

В истории развития теории автоматического управления четко выделяются три этапа. По классификации, приведенной в [40], первым был этап классической детерминированной теории автоматического регулирования, охвативший период времени с конца XIX по 40-е годы XX века. В этот период основными задачами управления были задача устойчивости и задача о качестве переходных процессов. Второй этап теории управления начался в 40—50-х годах нашего века и длился примерно до середины 70-х годов. Это — этап классической стохастической теории автоматического регулирования. Он характеризуется новой постановкой основной задачи теории управления: учесть случайные возмущения, действующие на систему, и обеспечить хорошую работу в условиях постоянно действующих помех.

Около 25-и лет назад в развитии теории автоматического управления начался новый этап, связанный с адаптивной постановкой основной задачи управления. Ее особенность состоит в отсутствии изначальных знаний о математической модели объекта управления, будь то

дифференциальные уравнения или плотности вероятностей случайных внешних воздействий. Объект — это черный ящик, подвергающийся неизвестным случайным воздействиям. Нам доступны только его входы и выходы. Цель системы управления (СУ) состоит в том, чтобы уже в процессе функционирования определить закон регулирования, обеспечивающий оптимальное поведение объекта. Для решения этой задачи в дополнение к основному контуру в систему управления вводится контур адаптации (см. рис. 1.12).

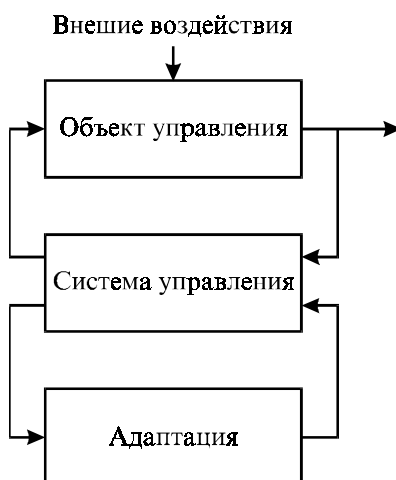


Рис. 1.12

Общая схема адаптивной системы управления

С самого начала третьего этапа огромное внимание уделялось адаптивному управлению линейными стационарными объектами с неизвестными параметрами (например, широко используемые методики, опирающиеся на построение наблюдателей). В рамках этого подхода в 80-х годах началось использование ИНС для решения задач управления. Полученные результаты показали, что ИНС представляют собой на просто новую методику в теории автоматического управления, а целую па-

радигму. Для нового направления в теории управления Вербосом было введено отдельное название — нейроуправление (neurocontrol) [41].

О цельности (парадигмоидальности) нейроуправления говорит то, что в нем, благодаря описанным выше свойствам ИНС, общим для различных нелинейных динамических объектов образом решаются задачи идентификации, синтеза систем управления, их анализа и аппаратной реализации. Результаты, полученные с применением ИНС в рамках адаптивной постановки основной задачи теории управления, легко могут использоваться и классическими подходами.

На рис. 1.13 представлена некоторая динамическая система (объект управления). В адаптивной постановке объект управления описывается своей функциональной моделью:

$$P\{\mathbf{u}(t), \mathbf{y}(t)\}, \quad (1.18)$$

связывающей вектор входных воздействий $\mathbf{u}(t)$ с вектором выходных сигналов $\mathbf{y}(t)$. Такое описание берет свое начало от идеи «черного ящика» и не является портретом динамического поведения объекта, а отражает только его функциональные связи.

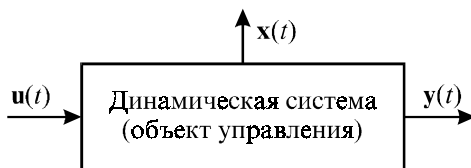


Рис. 1.13

Динамическая система

Введем векторы $\mathbf{u}(t)=(u_1(t), u_2(t), \dots, u_p(t))^T$, $\mathbf{x}(t)=(x_1(t), x_2(t), \dots, x_n(t))^T$ и $\mathbf{y}(t)=(y_1(t), y_2(t), \dots, y_m(t))^T$. Здесь $u_i(t)$ и $y_i(t)$ — вход и выход системы, соответственно, $x_i(t)$ — переменная состояния системы, p — размерность входного пространства, m — размерность выходного пространства, а n — порядок системы.

Классически, динамика такой системы описывается системой дифференциальных уравнений:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \Phi(\mathbf{x}(t), \mathbf{u}(t)), & t \in \mathfrak{R}^+; \\ \mathbf{y}(t) &= F(\mathbf{x}(t)). \end{aligned} \quad (1.19)$$

Здесь вектора функции $\Phi=(\phi_1, \phi_2, \dots, \phi_n)$ и $F=(f_1, f_2, \dots, f_m)$ — статические нелинейные преобразования. $\Phi: \mathfrak{R}^n \times \mathfrak{R}^p \rightarrow \mathfrak{R}^n$, $F: \mathfrak{R}^n \rightarrow \mathfrak{R}^m$. Вектор $\mathbf{x}(t)$ описывает состояние системы в момент времени t . Он определяется состоянием системы в начальный момент $t_0 < t$ и входом \mathbf{u} , определенном на интервале $[t_0, t)$. Выход системы $\mathbf{y}(t)$ полностью определяется состоянием системы \mathbf{x} в момент t .

Другим подходом является дискретное описание динамической системы. Введем разбиение по времени t_0, t_1, t_2, \dots , где $t_{i+1} = t_i + \Delta t$, и обозначим $\mathbf{x}(t_k)$, $\mathbf{y}(t_k)$ и $\mathbf{u}(t_k)$ как $\mathbf{x}(k)$, $\mathbf{y}(k)$ и $\mathbf{u}(k)$, соответственно. Тогда динамику системы можно описать следующими разностными уравнениями:

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi(\mathbf{x}(k), \mathbf{u}(k)); \\ \mathbf{y}(k) &= F(\mathbf{x}(k)). \end{aligned} \quad (1.20)$$

Здесь Φ и F аналогичны преобразованиям в (1.19).

Уравнения (1.19, 1.20) представляют динамику системы, как преобразование вход-выход. Для широкого класса задач эти формы представления равнозначны и могут быть сведены друг к другу. Однако, при описании систем управления с нейросетевыми элементами более удобным является дискретное представление объекта управления, поэтому в дальнейшем будем рассматривать динамические, представленные в виде (1.20). Тем не менее, результаты, полученные для дискретного описания, могут быть распространены и на непрерывный случай.

1.6.1. Идентификация.

Задача идентификации является фундаментальной в теории систем и, в частности, теории автоматического управления. Целью идентификации является построение идентификационной модели \hat{P} (см. рис. 1.14), аппроксимирующей объект P :

$$\|\hat{y} - y\| = \|\hat{P}(\mathbf{u}) - P(\mathbf{u})\| \leq \varepsilon, \quad \mathbf{u} \in U, \quad (1.21)$$

для некоторого заданного $\varepsilon > 0$ и определенной нормы $\|\cdot\|$. Здесь $\hat{y} = \hat{P}(\mathbf{u})$ — выход идентификационной модели, U — допустимое множество управления. Причем, как для статической, так и для динамической систем оператор P неявно определен парами сигналов вход-выход $\{\mathbf{u}, y\}$.

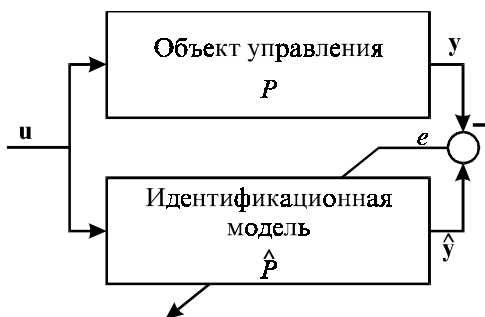


Рис. 1.14

Идентификация объекта управления

Выбор класса, к которому принадлежит оператор \hat{P} , и самого оператора определяется множеством факторов, связанных с требуемой точностью и аналитической трактуемостью модели. К ним относятся адекватность представления P с помощью \hat{P} , сложность идентификации, простота модели, возможность ее расширения и дополнения, а также возможность использования модели в реальном масштабе време-

ни. Выбор \hat{P} зависит и от имеющейся априорной информацией о структуре объекта.

Благодаря своим универсальным аппроксимирующим свойствам, ИНС представляют собой мощный инструмент для решения задачи идентификации нелинейных статических и динамических объектов управления. Основанные на ИНС дискретные идентификационные модели называются нейроэмуляторами (НЭ) или предикторами. В общем виде они описываются следующим нелинейным уравнением:

$$\hat{\mathbf{y}}(k+1) = NN(\hat{\mathbf{y}}(k), \hat{\mathbf{y}}(k-1), \dots, \hat{\mathbf{y}}(k-l_1), \mathbf{u}(k), \mathbf{u}(k-1), \dots, \mathbf{u}(k-l_2)), (1.22)$$

где $NN(\cdot)$ — преобразование вход-выход, выполняемое ИНС,

l_1 — глубина задержки обратной связи по выходу НЭ,

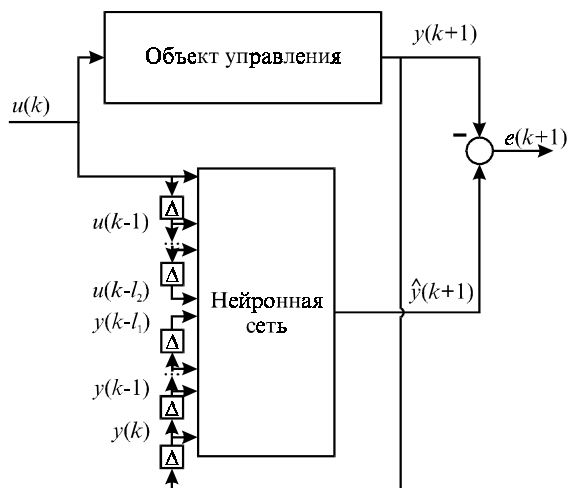
l_2 — глубина задержки по входу НЭ.

Различают одношаговые и краткосрочные предикторы (эмуляторы) (см. рис. 1.15). Одношаговый предиктор осуществляет предсказание выходного вектора объекта по его предыстории на один шаг вперед:

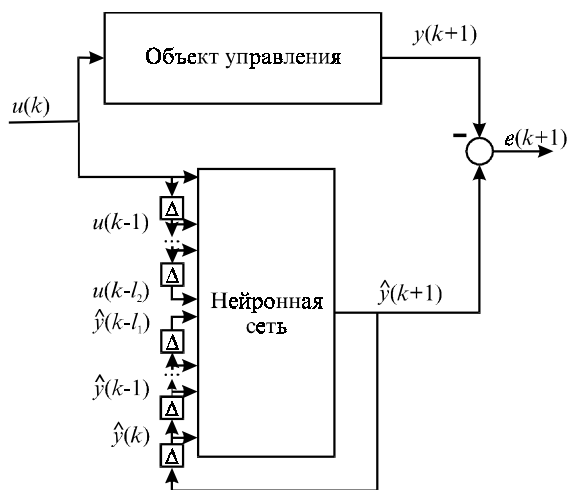
$$\hat{\mathbf{y}}(k+1) = NN(\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(k-l_1), \mathbf{u}(k), \mathbf{u}(k-1), \dots, \mathbf{u}(k-l_2)). (1.23)$$

Такие модели объектов не являются полными и используются только для прогнозирования поведения сложных систем.

Краткосрочные предикторы или просто нейроэмуляторы (1.22) являются полной моделью объекта в том смысле, что они могут использоваться независимо от самого объекта. С их помощью можно строить прогноз на несколько шагов вперед, отталкиваясь от сделанных ранее предсказаний. Тем не менее, само название краткосрочных предикторов подчеркивает, что глубина их предсказания является ограниченной. Нарастание ошибок, вызванных неточностью аппроксимации, со временем приводит к полному расхождению объекта и модели.



а)



б)

Рис. 1.15

Типы нейросетевых идентификационных моделей.

а) — одношаговый предиктор; б) — нейроэмулятор

Долгосрочный предиктор представляет собой предельный случай, который может быть получен лишь в том случае, если идентифика-

ция объекта проводилась с учетом всех возможных его состояний, и ошибка аппроксимации равна нулю.

Классически идентификация проводится на основании определенных гипотез и теоретических исследований по имеющимся экспериментальным данным. При этом особое внимание уделяется физическому смыслу полученной модели. Конечно, полученная с помощью ИНС идентификационная модель мало пригодна для дальнейшего анализа с целью выяснить суть происходящих внутри объекта процессов. Это связано с распределенностью проводимого ею преобразования вход-выход по всем нейронам сети. Однако отражение физической сути процессов и не является необходимым условием идентификации, так как в адаптивной постановке задача состоит в другом. Цель идентификации — оперативное построение прогноза поведения объекта при определенной стратегии управления.

1.6.2. Управление.

Предмет теории управления составляют анализ и синтез динамических систем, в которых изменение одной или нескольких переменных ограничивается в определенных пределах. Если функции Φ и F в (1.20) известны, то задачей управления является синтез контроллера, который формирует желаемый вход $\mathbf{u}(k)$, основываясь на информации, доступной в момент времени t_k .

В классической теории управления имеется множество хорошо разработанных методов синтеза контроллеров в частотной и временной областях для линейных систем типа:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k); \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned}\tag{1.24}$$

с известными матрицами $\mathbf{A}(n,n)$, $\mathbf{B}(n,p)$, $\mathbf{C}(m,n)$. Однако, для случая нелинейных систем, даже если $\Phi(\cdot)$ и $F(\cdot)$ определены, единых подходов с гарантированным результатом не существует.

Использование ИНС позволяет решать задачу управления нелинейным объектом путем создания адаптивной СУ с обучаемым нейроконтроллером (НК). Здесь под обучением подразумевается процесс выработки в СУ желаемой реакции на внешние сигналы путем многократных воздействий на систему и внешней корректировки. Внешняя корректировка осуществляется «учителем», которому известна желаемая реакция СУ на определенные воздействия. Таким образом, при обучении «учитель» сообщает системе дополнительную информацию о том, верна или неверна ее реакция (см. рис. 1.16).

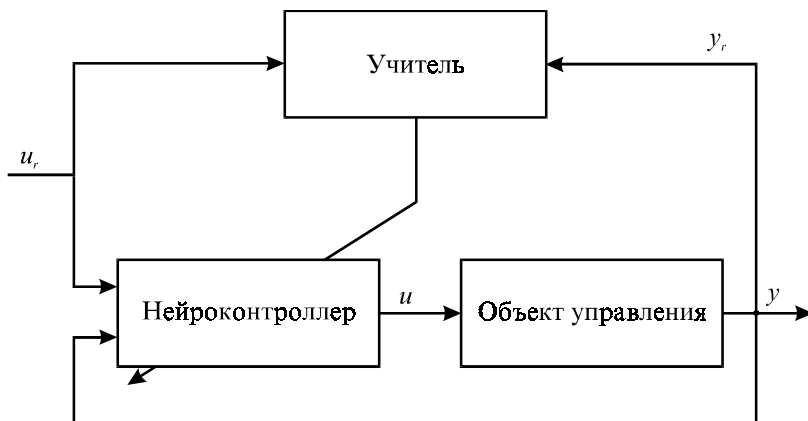


Рис. 1.16

Общая схема системы управления с обучаемым нейроконтроллером

При адаптации обучение используется для получения информации о состоянии и характеристиках СУ, необходимой для оптимального управления в условиях неопределенности. По существу, адаптацию можно отождествить с оптимизацией в условиях недостаточной априорной информации [42].

Существует множество подходов к применению ИНС в качестве НК. Так в работе [43] предлагается использовать ИНС, как нелинейные усилители при интегральной, дифференциальной и пропорциональной частях ПИД-контроллера (рис. 1.17).

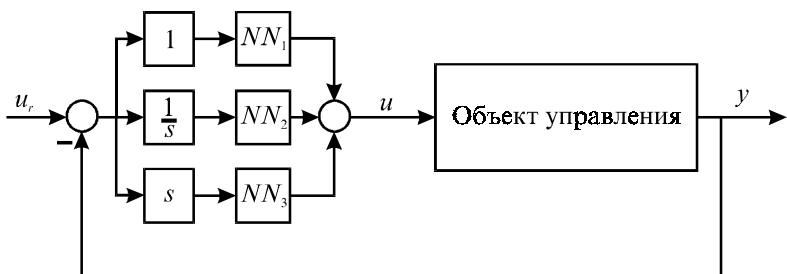


Рис. 1.17

Система управления с комбинированным ПИД-нейроконтроллером

В тоже время в [44] (одной из первых статей по нейроуправлению) используется дискретная СУ с НК, получающим на вход сигнал управления системой $\mathbf{u}_r(k)$ и задержанные несколько раз сигналы с выхода объекта управления. В общем случае НК такого типа может получать и задержанные сигналы со своего выхода (рис. 1.18). Таким образом, он формирует управляющее воздействие по следующему закону:

$$\mathbf{u}(k) = NN(\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(k-l_1), \mathbf{u}(k-1), \mathbf{u}(k-2), \dots, \mathbf{u}(k-l_2), \mathbf{u}_r(k)), \quad (1.25)$$

где l_1 и l_2 — глубины задержек обратных связей по выходу и входу объекта управления, соответственно.

Если первый тип СУ с ИНС является компромиссным решением в рамках классического подхода, то второй тип построен только на ИНС.

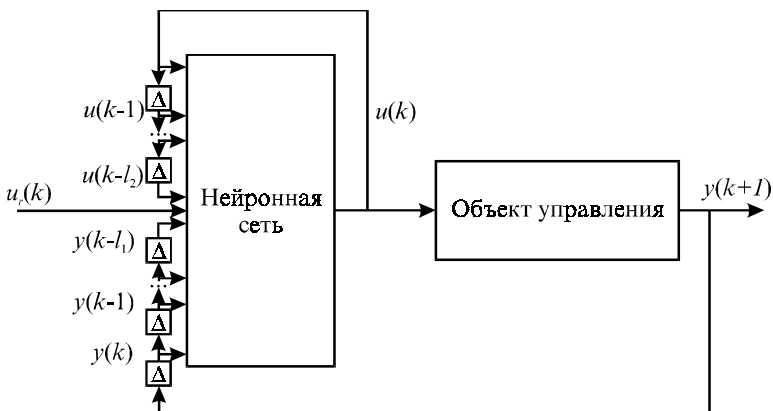


Рис. 1.18

Система управления с нейроконтроллером

СУ с ИНС можно классифицировать как адаптивные системы [45]. В этом случае выделяются замкнутые и разомкнутые адаптивные СУ. Под замкнутостью СУ понимается прямое использование входных и выходных сигналов объекта для адаптации управления, без промежуточной идентификации объекта. Наиболее часто используются замкнутые нейросетевые системы управления (НСУ) с эталонной моделью (рис. 1.19) [46].

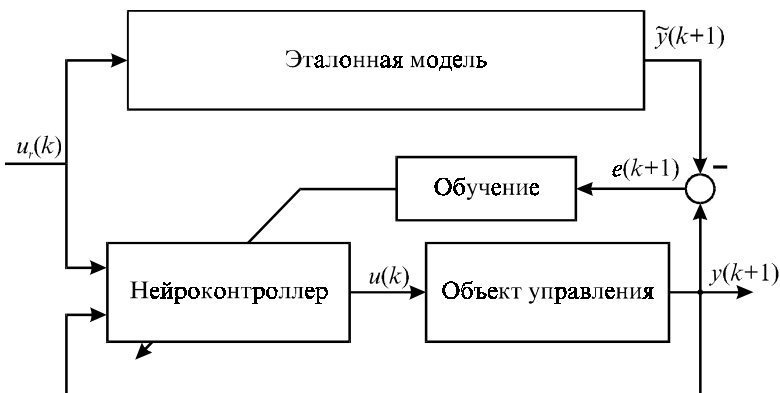


Рис. 1.19

Нейросетевая система управления с эталонной моделью

Пусть имеется определенная эталонная модель СУ:

$$R\{\mathbf{u}_r(t), \tilde{\mathbf{y}}(t)\}, \quad (1.26)$$

где $\mathbf{u}_r(t)$ — вектор входных сигналов системы,

$\tilde{\mathbf{y}}(t)$ — ее желаемый выход.

Тогда целью обучения НК является получение ограниченного управления $\mathbf{u}(t)$ для $t \geq 0$ такого, что

$$\lim_{t \rightarrow \infty} \|\tilde{\mathbf{y}}(t) - \mathbf{y}(t)\| = 0. \quad (1.27)$$

В этом случае объект будет отслеживать желаемую траекторию, определенную эталонной моделью [47]. В [48] показано, что возможно построение адаптивного НК на базе ИНС с контролируемым обучением, обеспечивающего СУ глобальную асимптотическую устойчивость.

Необходимым условием функционирования СУ с эталонной моделью является правильный выбор тестовых входных воздействий \mathbf{u}_r , так как от типа тестового сигнала зависит точность настройки основной системы на эталонную модель.

Другим распространенным типом являются замкнутые СУ без эталонной модели (рис. 1.20).

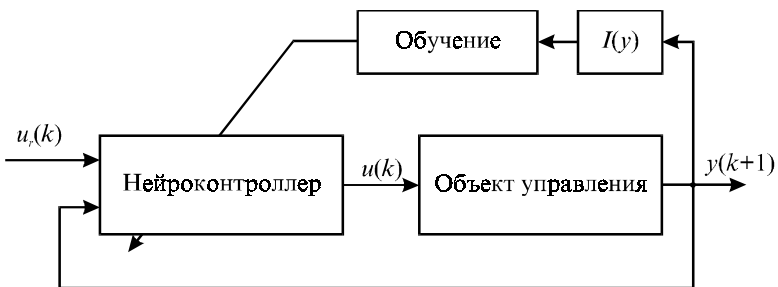


Рис. 1.20

Нейросетевая система управления с экстремальным законом управления

Целью обучения НК в них является получение закона управления, обеспечивающего заданное (обычно экстремальное) значение неко-

того функционала качества I , определяемого текущим состоянием системы.

$$\mathbf{u}(t) \Rightarrow I(\mathbf{y}(t)) \rightarrow \text{ext} . \quad (1.28)$$

Существуют также схемы разомкнутых адаптивных НСУ с НЭ объекта управления. В этом случае НЭ может использоваться либо для оценки требуемого сигнала управления при заданном выходном состоянии объекта (если НЭ описывает обратную динамику объекта управления [46]), либо для оценки значений недоступных непосредственному измерению параметров объекта α [49, 50]. Эти данные используются затем для обучения НК (см. рис. 1.21).

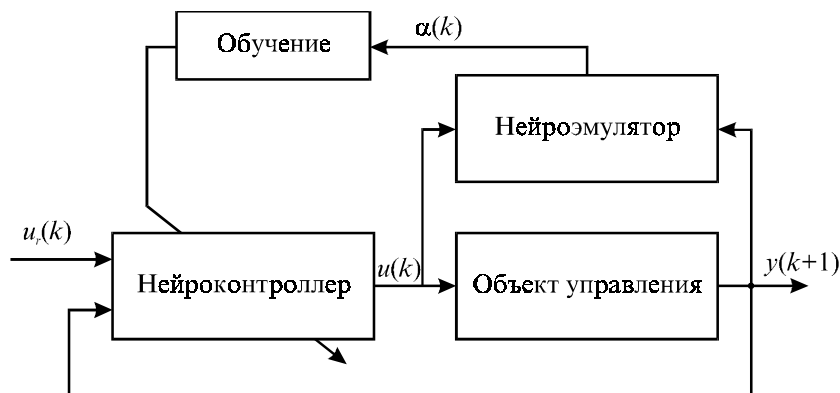


Рис. 1.21

Нейросетевая система управления с нейроэмулятором

1.7. Постановка задачи на исследование

Целью диссертационной работы является *разработка методик эволюционного синтеза нейросетевых компонентов систем автоматического управления, основывающихся на применении генетических алгоритмов.*

Для этого необходимо решить следующие задачи:

1. Адаптировать классическую схему генетического алгоритма для синтеза на базе прямонаправленных искусственных нейронных сетей идентификационной модели динамического объекта;
2. Оценить перспективы применения идеологии задержанного стимулируемого обучения для тренировки компонентов нейросетевой системы управления с помощью генетического алгоритма;
3. Определить достаточную систему типовых входных воздействий, используемых для обучения нейросетевых компонентов системы управления;
4. Оценить верхнюю границу размера скрытого слоя трехслойных нейронных сетей, используемых для решения задач идентификации и управления. Разработать методику определения структуры нейросетевых идентификационных моделей и нейроконтроллеров;
5. Оценить параметрическую и структурную чувствительности, помехоустойчивость и надежность синтезируемых с помощью генетического алгоритма нейроконтроллеров;
6. Разработать методику синтеза нейроконтроллера по нейросетевой модели динамического объекта управления, оценить точность и эффективность полученного решения.

ГЛАВА 2

РАЗРАБОТКА МЕТОДИКИ СИНТЕЗА НЕЙРОЭМУЛЯТОРОВ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

Идентификация с помощью ИНС, так же как и в рамках классических моделей, проводится по данным вход-выход, полученным в ходе экспериментов с объектом управления. Объединив эти данные, можно создать полный тренировочный набор шаблонов, а значит, естественным выбором метода настройки для НЭ является прямое контролируемое обучение (см. рис. 1.8).

Рассмотрим нейросетевую идентификационную модель (1.22) динамического объекта вида (1.20). Для ее обучения необходим тренировочный набор шаблонов:

$$\Xi\{z_i, d_i\}_{i=1, N}, \quad z \in \mathfrak{R}^g, \quad d \in \mathfrak{R}^m \quad (2.1)$$

где z — входной шаблон,

d — выходной шаблон,

N — количество шаблонов,

$g=(l_1+1)+(l_2+1)$ — размер входного слоя сети.

Вектор входных сигналов сети z включает в себя:

$$z = (y(k), y(k-1), \dots, y(k-l_1), u(k), u(k-1), \dots, u(k-l_2)), \quad (2.2)$$

вектор желаемых выходов сети равен:

$$d = y(k+1). \quad (2.3)$$

Глубины задержек l_1 и l_2 , размер тренировочного набора N и типы входящих в него шаблонов определяются такими, чтобы обученный по Ξ НЭ удовлетворял неравенству (1.21). Причем если выбор первых трех

параметров ничем не ограничен, то последний параметр — тип шаблонов — во многих случаях невозможно полностью контролировать.

Наиболее типичным и удобным случаем идентификации объекта управления является моделирование в лабораторных условиях. В ходе экспериментов исследователь может сформировать группу тестовых входных воздействий, которые позволят объединить в тренировочный набор шаблонов данные, наиболее полно представляющие свойства объекта. При этом могут использоваться входные воздействия и состояния идентифицируемого объекта, которые никогда не возникают при его нормальной эксплуатации.

Другой ситуацией является идентификация объекта управления в процессе его нормальной эксплуатации. Такая задача типична для крупных динамических систем, образующихся при объединении нескольких более простых объектов в единый технологический комплекс. Проведение экспериментов с ними часто невозможно или сопряжено с большими дополнительными затратами. Тренировочный набор шаблонов в этом случае формируется по данным об объекте, собранным во всех допустимых режимах эксплуатации. Так как повлиять на репрезентативность такого набора нельзя, то целью исследователя становится выбор параметров N , l_1 и l_2 , обеспечивающих наилучшую аппроксимацию имеющихся данных.

Уникальные качества идентификационных нейросетевых моделей наиболее полно проявляются именно во втором случае, так как они позволяют синтезировать НЭ объекта, не используя знаний о его структуре.

При решении задач идентификации обычно применялись многослойные перцептроны. Это было связано, прежде всего, с возможностью их прямого контролируемого обучения с помощью популярного алгоритма обратного распространения ошибки. Однако в ходе исследо-

ваний последних 5-и лет было показано, что для решения этой задачи лучше подходят РБФ-сети. Они обеспечивают более быстрый синтез НЭ и позволяют контролировать в процессе обучения уровень обобщения в сети. Последнее очень ценно, так как от качества обобщения зависят все остальные свойства НЭ.

2.1. РБФ-сети

Традиционно архитектура ИНС, методы обучения и теоретические обоснования их свойств строились на основе модели нейрона Мак-Каллока-Питтса. Однако в конце 80-х годов была предложена новая схема трехслойной ИНС, использующая другую модель нейрона скрытого слоя (1.13). РБФ-сеть появилась как нелинейная модификация перцептрона Розенблатта, использующая оригинальный метод обучения.

Как и многослойный перцептрон РБФ-сеть является универсальным аппроксиматором [51], поэтому любому многослойному перцептрону можно сопоставить РБФ-сеть, осуществляющую аналогичное преобразование, и наоборот.

Впервые радиально-базисные (потенциальные) функции были введены при решении многомерных интерполяционных задач, и сейчас являются одной из главных областей теоретических исследований в численном анализе. В работе [26] они были использованы для построения ИНС, обладающей гарантированным качеством обобщения новых данных.

По сути, обучение РБФ-сети является аппроксимационной задачей нахождения многомерной поверхности, обеспечивающей наилучшее совпадение с набором тренировочных данных. В отличие от многослойных перцептронов, которые глобально аппроксимируют нелиней-

ное преобразование вход-выход, РБФ-сети строят локальную аппроксимацию.

Из-за экспоненциального убывания активационных функций нейронов скрытого слоя выход РБФ-сети для входных данных, лежащих вдали от тренировочных шаблонов, стремится к нулю. Такая локальность обеспечивает быстрое обучение сети и снижает типичную для многослойных перцептронов зависимость качества обучения от последовательности подачи тренировочных данных. С другой стороны, это приводит к необходимости использования большего числа скрытых нейронов.

Архитектура РБФ-сети включает три слоя нейронов (см. рис. 2.1). Первый слой служит для ретрансляции входных сигналов. Нейроны скрытого слоя осуществляют нелинейное преобразование входных сигналов. Линейные выходные нейроны суммируют сигналы со скрытого слоя и формируют выход сети, при этом сдвиг полагается равным нулю. В отличие от многослойного перцептрона, для скрытых нейронов РБФ-сети отсутствует понятие весов синаптических связей. Целью обучения является настройка параметров радиально-базисных активационных функций скрытых нейронов и весов выходных нейронов.

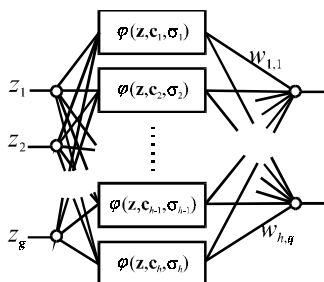


Рис. 2.1

Архитектура $g-h-q$ РБФ-сети

Методика обучения РБФ-сети с активационными функциями нейронов скрытого слоя типа (1.13) основывается на процедуре строгой интерполяции и заключается в следующем [52]:

Для данного набора тренировочных шаблонов $\Xi\{\mathbf{z}_i, d_i\}_{i=1, \overline{N}}$,

$\mathbf{z} \in \mathfrak{R}^g$, $d \in \mathfrak{R}^1$ необходимо найти такие значения параметров σ_j , w_j , и векторов \mathbf{c}_j , $j=1, \dots, N$, что функция

$$F(\mathbf{z}) = \sum_{j=1}^N w_j e^{-\frac{\|\mathbf{z}-\mathbf{c}_j\|^2}{\sigma_j^2}} \quad (2.4)$$

удовлетворяет интерполяционному условию

$$F(\mathbf{z}_i) = d_i, \quad i = \overline{1, N}. \quad (2.5)$$

Определим размер выходного слоя сети q равным 1. Размер скрытого слоя положим равным количеству тренировочных шаблонов $h=N$ и выберем входные шаблоны из тренировочного набора в качестве центров радиально-базисных активационных функций:

$$\mathbf{c}_j \cong \mathbf{z}_j, \quad j = \overline{1, N}. \quad (2.6)$$

Значения ширины окон σ_j будем считать определенными из условий, которые будут оговорены ниже.

Тогда, подставляя (2.4) в (2.5), получим набор линейных уравнений относительно весов w_j , который может быть представлен в матричном виде:

$$\mathfrak{G}\mathbf{w} = \mathbf{D}. \quad (2.7)$$

Здесь

$$\mathfrak{G} = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{pmatrix}, \quad (2.8)$$

и

$$\varphi_{ij} = e^{-\frac{\|z_i - z_j\|^2}{\sigma_j^2}} \quad (2.9)$$

Квадратная матрица $\mathfrak{G}(N, N)$ называется интерполяционной матрицей, \mathbf{w} — вектор синаптических весов, \mathbf{D} — вектор желаемых выходов.

Как указано в [30], для радиально-базисных функций $\varphi(\cdot)$ интерполяционная матрица \mathfrak{G} является положительно определенной. Поэтому возможно решить (2.7) относительно \mathbf{w} и получить:

$$\mathbf{w} = \mathfrak{G}^{-1} \mathbf{D} \quad (2.10)$$

Решение (2.10) обеспечивает прохождение аппроксимационной поверхности через все точки тренировочного набора шаблонов. Определение \mathbf{w} не составляет трудностей, если интерполяционная матрица неособенная, однако это условие часто нарушается при решении практических задач.

Уравнение (2.7) описывает случай РБФ-сети с одним выходным нейроном, однако эта методика может быть применена и для многомерного выходного пространства $\mathbf{d} \in \mathfrak{R}^m$, $m > 1$. В этом случае для каждой проекции выходных шаблонов d_i , $i = \overline{1, m}$ последовательно записывается уравнение (2.7), и из (2.10) определяется вектор синаптических весов \mathbf{w} соответствующего выходного нейрона сети.

Данная методика никак не определяет ширины окон активационных функций σ_j , оставляя выбор за исследователем [53]. Значение этих параметров влияет на гладкость преобразования, осуществляемого сетью, и должно быть достаточно большим. В случае слишком узких окон аппроксимационная поверхность превращается в набор холмов, вершины которых совпадают с выходными шаблонами. При чрезмерно широких — интерполяционная матрица \mathfrak{G} становится особенной, а решение (2.10) невозможным. Для гарантированного получения требуемого уровня обобщения в РБФ-сети описанную методику обучения необхо-

димо дополнить поисковой процедурой выбора параметров σ_j . Хотя для задач небольшой размерности этот выбор может осуществляться и вручную.

В общем, следует заметить, что структура РБФ-сети требует использования значительного размера скрытого слоя. Этому существует формальное объяснение в рамках задачи классификации. РБФ-сеть нелинейно отображает сигналы из входного пространства в пространство выходов нейронов скрытого слоя (скрытое пространство). Выходные нейроны являются линейными, а поэтому, согласно теореме Ковера [30], все входные шаблоны могут быть линейно разделены лишь при достаточно высокой размерности скрытого пространства. Это условие определяет главный недостаток РБФ-сетей — *проклятие размерности*, так как с увеличением размерности входного пространства число необходимых нейронов скрытого слоя возрастает экспоненциально.

2.2. Генетический алгоритм

Генетические алгоритмы являются методами глобальной оптимизации, которые базируются на теоретических достижениях синтетической теории эволюции и современной генетики. Первые работы, посвященные ГА, появились в середине 60-х годов, и с тех пор их развитию и применению уделяется большое внимание [32]. С середины 80-х ГА стали применять для синтеза ИНС различного типа.

Главным достоинством ГА является возможность получения с его помощью глобального решения. Задача обучения ИНС, как и большинство реальных задач проектирования, является многоэкстремальной, поэтому все классические методы оптимизации, основывающиеся на использовании градиента целевой функции, не могут гарантировать нахождение наилучшего решения [54]. Многочисленные исследования,

проводившиеся в течение 30-и лет, доказали, что ГА является лучшим из существующих методом, обеспечивающим решение многомерных многоэкстремальных задач глобальной оптимизации [55]. Сегодня на базе персонального компьютера с помощью ГА возможно выбрать наилучшее решение среди 10^6 — 10^7 локальных экстремумов в поисковом пространстве размерностью порядка 10^3 — 10^4 .

Другой особенностью ГА является его высокая эффективность при работе с нестационарным окружением [56]. Хромосомное представление информации, а также поиск решения с помощью популяции поисковых точек позволяют ГА находить глобальный оптимум сильно зашумленной или быстро меняющейся целевой функции [57]. При этом в работу алгоритма не вносятся никаких изменений по сравнению со стационарным случаем, и скорость поиска решения сохраняется на том же уровне.

Методологическая основа ГА опирается на гипотезу селекции, которая в самом общем виде может быть сформулирована так [52]:

Чем выше приспособленность особи, тем выше вероятность того, что в потомстве, полученном с ее участием, признаки, определяющие приспособленность, будут выражены еще сильнее.

В рамках ГА задача оптимизации решается в ходе эволюции искусственной популяции, особи которой представляют собой отдельные варианты решения [58]. На каждом этапе эволюции в популяции проводится селекция, то есть отбираются худшие особи, которые погибают, и лучшие, которые дают потомство. Приспособленность особи определяется с помощью целевой функции. Новые особи, полученные в результате скрещивания родительских, замещают погибших, и процедура селекции повторяется. В течение ряда эпох (этапов) эволюции популяция адаптируется к внешнему окружению, в качестве которого выступает целевая функция задачи оптимизации, и все особи

целевая функция задачи оптимизации, и все особи оказываются одинаково приспособленными. Набор параметров оптимизации, описываемый лучшей особью популяции, представляет собой глобальное решение.

Рассмотрим решение задачи нелинейного программирования с помощью описанной в [52] версии ГА:

$$f(\chi^*) = \min_{\chi \in X} f(\chi), \quad (2.11)$$

где $\chi \in \mathbb{R}^l$ — вектор координат поисковой точки,

χ^* — оптимальное решение,

X — область поиска.

Для простоты будем считать, что все возможные ограничения в виде равенств или неравенств уже учтены в целевой функции f в виде свертки критериев, с помощью метода штрафных функций или другим образом.

Возможное решение задачи (2.11) представляется в ГА в виде особи $A\{\chi, f(\chi)\}$. Она включает в себя вектор $\chi(\chi_1, \chi_2, \dots, \chi_l)$, закодированный в l линейных хромосомах (одна на каждую координату), а также соответствующее значение целевой функции $f(\chi)$.

Классическая версия ГА является гаплоидной, то есть каждое решение однозначно описывается единственным набором хромосом. Однако такая схема подвержена преждевременной сходимости, что не всегда позволяет найти глобальный оптимум. Более удачным является диплоидное представление информации в особи. При этом вектор χ описывается двойным набором хромосом, доставшимся от двух родителей. Каждый отдельный набор хромосом носит название гаметы, двойной набор называется зиготой.

Хромосома представляет собой множество генов, расположенных в отдельных локусах. Количество локусов определяет длину хромосомы, каждый локус может быть занят разными генами. Для кодирования

информации в хромосоме в рамках диплоидной версии ГА применяется четырехбуквенный алфавит: {RrDd} [52] (см. рис. 2.2). Здесь ген R означает рецессивное 1, ген r — рецессивное 0, ген D — доминантное 1, а ген d — доминантное 0. 1 и 0 соответствуют двоичному коду Грея. Чтобы декодировать значение вектора решения χ , вначале по двойному набору хромосом с учетом соотношений доминирования определяется с его двоичное представление в коде Грея.

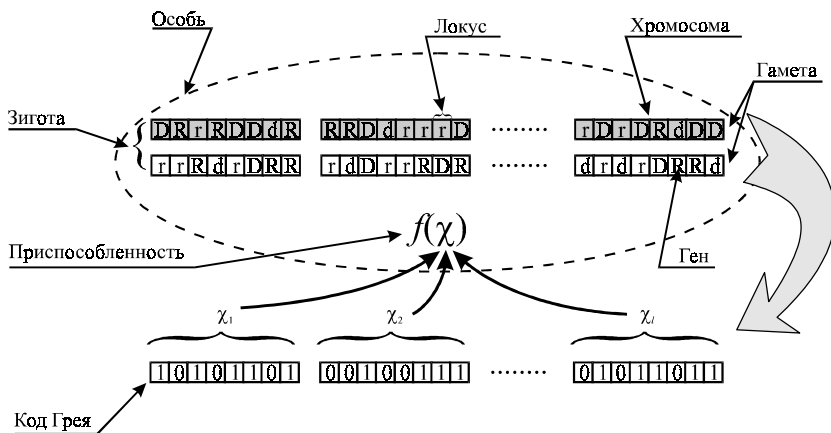


Рис. 2.2
Особь

При определении двоичного представления вектора x используется таблица доминирования (см. рис. 2.3) [53], позволяющая определить, что кодирует каждая пара генов R_d, DD, rR, ..., содержащаяся в гомологичных локусах двух хромосом (то есть имеющих одинаковый порядковый номер). Чтобы однозначно разрешить вопрос в конфликтных ситуациях Dd, dD, Rr, и rR (затемненные ячейки на рис. 2.3) во внимание принимается, от какой родительской особи достались гены, боле слабой или более сильной.

		Гены более приспособленной особи			
		D	d	R	r
Гены менее приспособленной особи	D	1	0	1	1
	d	1	0	0	0
	R	1	0	1	1
	r	1	0	0	0

Рис. 2.3

Таблица доминирования

В отличие от методов, использующих вещественное представление поисковой точки, ГА осуществляет поиск на v -мерном гиперкубе, заданном на множестве $\{0,1\}$, где $v = \sum_{i=1}^l \lambda_i$, λ_i — длина i -й хромосомы.

Это означает, что область поиска X заменяется охватывающей ее прямоугольной областью: X_r :

$$X_r: \left\{ \sup \chi_i^1 = \sup \chi_i^2, \inf \chi_i^1 = \inf \chi_i^2 \mid \chi^1 \in X, \chi^2 \in X_r, i = \overline{1, l} \right\}, \quad (2.12)$$

которая дискретизирована по каждой из проекций с шагом η_i равным:

$$\eta_i = \frac{\sup \chi_i - \inf \chi_i}{2^{\lambda_i} - 1}, \quad (2.13)$$

где $\chi \in X$, $i = \overline{1, l}$ — номер проекции.

ГА по определенным правилам перебирает вершины гиперкуба, перемещая тем самым поисковые точки в пространстве X_r . Величина шага η_i определяет предельно достижимую точность определения координат решения χ^* . Методы, работающие с вещественным представлением, лишены такого ограничения, однако для большинства практических задач этот недостаток может быть устранен путем выбора достаточно большого λ_i по всем координатам.

В качестве метрики в двоичном пространстве поискового гиперкуба используется хэммингово расстояние, то есть расстояние между

двумя бинарными цепочками, описывающими координаты вершины гиперкуба, определяется количеством разных бит в них. Использование кода Грея обеспечивает непрерывность преобразования вещественного представления поисковой точки в двоичные координаты вершины гиперкуба. То есть двум соседним точкам вещественного пространства, отстоящим по одной из координат на шаг η_i , соответствуют вершины, принадлежащие одной грани, а значит хэммингово расстояние между ними равно 1. Это важное условие не выполняется при использовании простого двоично-десятичного кода [59].

Полученные с помощью таблицы доминирования n двоичных чисел определяют координаты вершины поискового гиперкуба. Каждое из них преобразуется в натуральное число k_i , по которому вычисляется вещественное значение соответствующей координаты решения χ_i согласно формуле:

$$\chi_i = k_i \eta_i + \gamma_i, \quad (2.14)$$

$$\gamma_i = \inf \chi_i, \chi \in X, i = \overline{1, l}. \quad (2.15)$$

Для декодированного из хромосом вектора χ определяется значение целевой функции $f(\chi)$. Таким образом, особь приобретает оценку ее приспособленности.

Столь сложное описание поисковой точки в виде особи необходимо для реализации принципа коллективного поиска, заимствованного ГА из биологии. В отличие от большинства классических методов, ГА работает не с одной поисковой точкой, а с их популяцией (этот принцип в определенной мере используют также метод деформируемого многогранника [60] и метод группового учета аргумента [61]). Это, а также генетический способ передачи информации в череде поколений обеспечивает высокую поисковую эффективность ГА.

Основным элементом ГА является популяция особей $\Pi\{A_i\}_{i=1,\mu}$.

Размер популяции μ является одним из главных факторов, определяющих эффективность работы всего алгоритма. Чем больше популяция, тем выше ее поисковая способность [62]. Однако на практике величина μ ограничена вычислительной мощностью используемых компьютеров и лежит в пределе от 10 до 500 особей.

На каждом этапе работы алгоритма множество образующих популяцию Π особей упорядочивается по возрастанию в соответствии с величиной целевой функции f . В отсортированной популяции выделяются и элиминируются $\rho \times \mu$ последних особей, обладающих наихудшей приспособленностью, то есть наибольшими значениями целевой функции (для задачи минимизации). Здесь ρ — коэффициент давления отбора, обычно принимаемый равным 0,1 [63]. Оставшиеся $(1-\rho) \times \mu$ особей формируют родительскую группу. Таким образом, в популяции проводится антиотбор, направленный против худших решений. Из особей, входящих в родительскую группу, произвольно с равной вероятностью выбирается $\rho \mu$ пар родителей. Потомки, полученные от каждой пары, занимают место $\rho \mu$ худших особей, тем самым, поддерживая постоянной общую численность популяции.

Генерирование новой особи происходит в несколько этапов (на рис. 2.4 этот процесс проиллюстрирован для случая одномерной задачи $l=1$). Вначале каждая из особей родителей должна произвести гамету — одинарный набор хромосом, пара которых составит зиготу потомка. Для этого оба набора хромосом родителя копируются, и внутри копии проводится кроссовер, то есть каждая из l пар гомологичных хромосом рассекается в произвольном месте и обменивается своими участками (см. рис. 2.4), сохраняя при этом порядок следования генов.

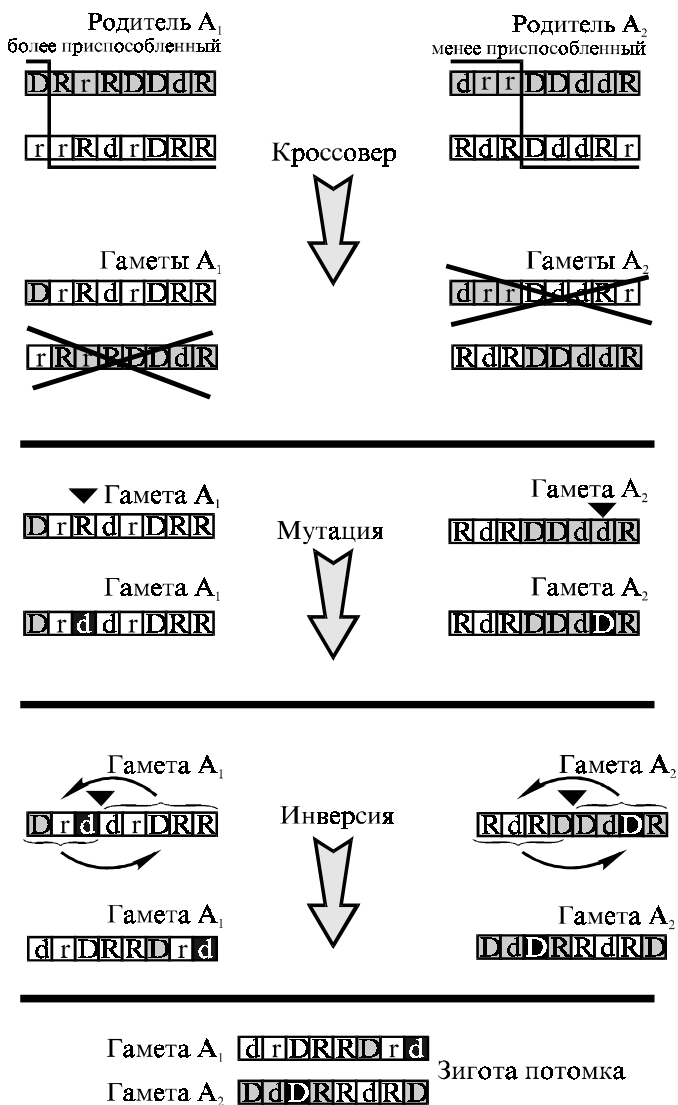


Рис. 2.4

Процесс образования зиготы потомка

Количество таких операций расщепления и обмена является случайной величиной с равномерным законом распределения и, обычно, изменяется в диапазоне [0,3]. Кроме внутриврохромосомных обменов с той

же вероятностью проводится обмен самими хромосомами между двумя наборами. В результате кроссовера образуются две гаметы, одна из них равновероятно отбрасывается, другая остается.

На следующем этапе полученная гамета подвергается мутациям, то есть в произвольных локусах набора хромосом имеющиеся гены заменяются альтернативными (см. рис. 2.4). При четырехбуквенном алфавите, это — один из трех других генов. В процессе мутации гена меняется либо кодируемый знак (0 или 1), либо активность гена (доминантный или рецессивный), либо знак и активность одновременно. Вероятность мутации равная для всех генов набора хромосом и выбирается такой, чтобы в среднем измененным оказался каждый 500-й ген в популяции.

Последним изменением, происходящим в новой гамете, является операция инверсии (см. рис. 2.4). В ходе ее хромосомы рассекаются в произвольном месте и их части переставляются, изменяя порядок следования генов. Вероятность инверсии для каждой хромосомы гаметы обычно низка, и в данном случае выбирается равной 0,01 [64].

Все описанные генетические операции могут быть проиллюстрированы на поисковом гиперкубе. Так кроссовер генерирует потомка в вершинах, которые находятся на пересечении граней, соприкасающихся с порождающими вершинами. Мутация приводит к перемещению поисковой точки по одному из ребер гиперкуба, а инверсия — вызывает отражение точки по диагонали в плоскости одной из соприкасающихся граней. Тем не менее, применяемые в ГА кроссовер, мутация и инверсия не изобретены человеком, а заимствованы из природы. Именно эти три операции обеспечили успешность естественной эволюции.

Полученные от двух родителей гаметы объединяются вместе и образуют зиготу потомка. После декодирования вектора параметров и

оценки приспособленности новая особь занимает в популяции место одной из элиминированных.

На рис. 2.5 представлена блок-схема диплоидной версии ГА. В начале работы алгоритма создается популяция.

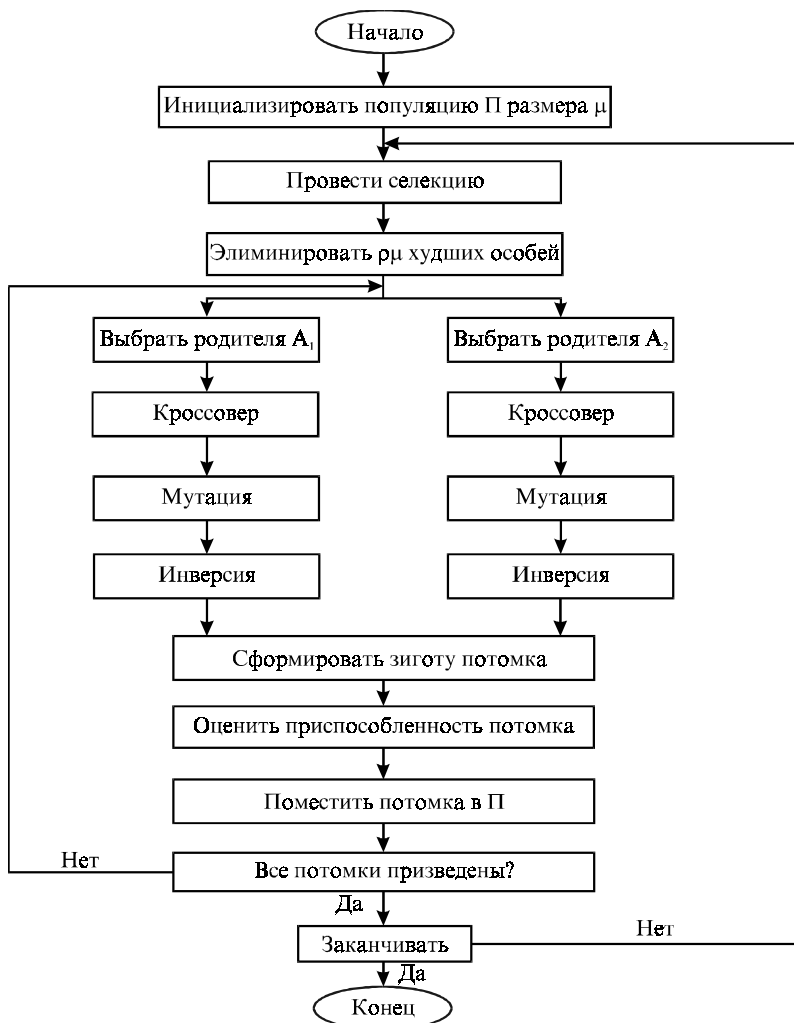


Рис. 2.5

Диплоидная схема репродуктивного плана Холланда

Двойной набор хромосом каждой особи может быть сгенерирован случайным образом, либо по уже имеющемуся вектору координат возможного решения. После проведения оценки приспособленности особь включается в популяцию. Сформированная популяция упорядочивается в соответствии с приспособленностью особей, и худшие из них элиминируются. Далее производятся потомки от случайно выбранных пар родителей, которые занимают место худших особей. Процесс итеративно повторяется, начиная с упорядочивания новой популяции.

Критерием остановки является разнообразие популяции, которое оценивается ее эффективным радиусом r_e :

$$r_e = \frac{1}{\mu - 1} \sum_{i=2}^{\mu} \rho(A_1, A_i), \quad (2.16)$$

$$\rho(A_1, A_i) = \sum_{j=1}^l (\chi_j^{A_1} - \chi_j^{A_i})^2, \quad (2.17)$$

где χ^{A_i} — вектор координат решения, описываемого особью A_i .

При нахождении глобального оптимума популяция сжимается в пространстве \mathcal{R}^l вблизи точки χ^* , и ее эффективный радиус r_e уменьшается. После достижения некоторого $r_e < \varepsilon$ поиск прекращается. Значение вектора χ^{A_1} лучшей особи популяции является решением задачи (2.11).

ГА является мощным поисковым средством. Однако его использование требует значительных вычислительных ресурсов и должно быть оправдано сложностью задачи. Несомненно, что любой классический метод будет эффективнее ГА в случае задачи низкой размерности, имеющей всего один экстремум. Однако большинство практических задач оптимизации, опирающихся на реальные данные, не отвечают таким требованиям. Именно для них и спроектирован ГА, чтобы быстро и точно найти лучшее решение там, где другие методы работать не могут.

2.3. Синтез РБФ-сети с помощью генетического алгоритма

Как было показано в подразделе 2.1 классическая методика обучения РБФ-сети имеет два недостатка. Во-первых, она не определяет правила выбора ширины окон σ_j активационных функций нейронов скрытого слоя, во-вторых, размер скрытого слоя должен быть равен числу тренировочных шаблонов, а значит может достигать больших значений. Последнее приводит к увеличению размерности интерполяционной матрицы и делает решение (2.10) вычислительно сложным. Использование ГА позволяет решить описанные проблемы [65], автоматизируя процесс выбора параметров скрытых нейронов при минимальном размере сети.

Известна методика эволюционного синтеза РБФ-сетей, использующая комбинацию ГА с методом обучения растущих сетей [66], однако ее недостатком является большой объем необходимых вычислений и высокая параметрическая чувствительность [67]. Последнее обстоятельство практически исключает возможность применения такой методики для обучения РБФ-сетей по данным, содержащим шумы.

В большинстве практических случаев используемый для синтеза РБФ-сети набор тренировочных шаблонов Ξ избыточен, то есть аппроксимационная поверхность может быть успешно построена по (2.4) и с меньшим, чем N количеством нейронов скрытого слоя. Однако заранее не известно, какая часть шаблонов является репрезентативной, а какая — избыточной. Поэтому решим задачу (2.4, 2.5) в другой постановке [53]:

Для данного набора тренировочных шаблонов $\Xi\{\mathbf{z}_i, \mathbf{d}_i\}_{i=1, N}$,

необходимо найти такую тренировочную выборку шаблонов

$\Xi_r\{\mathbf{c}_i, \mathbf{d}_i\}_{i=1, N_r}$, $N_r \leq N$ и значения параметров σ_j , что функция

$$F(\mathbf{z}) = \sum_{j=1}^{N_r} w_j e^{-\frac{\|\mathbf{z}-\mathbf{c}_j\|^2}{\sigma_j^2}} \quad (2.18)$$

удовлетворяет интерполяционному условию

$$F(\mathbf{z}_i) = \mathbf{d}_i, \quad i = \overline{1, N}. \quad (2.19)$$

Как и ранее, значения w_j , $j = \overline{1, N_r}$ вычисляются по (2.10), интерполяционная матрица \mathfrak{A} при этом имеет размер $N_r \times N_r$. Длину выборки N_r , входящие в нее шаблоны и ширину окон активационных функций σ_j , определим с помощью ГА следующим образом.

Объединим все неизвестные из задачи (2.18, 2.19) в вектор проектирования:

$$\chi = \left(\sigma_1, \mathbf{c}_1, \sigma_2, \mathbf{c}_2, \dots, \sigma_{N_m}, \mathbf{c}_{N_m} \right), \quad (2.20)$$

где \mathbf{c}_j — вектор координат центра активационной функции j -го нейрона,

$N_m \leq N$ — максимально возможный размер тренировочной выборки, и следовательно, скрытого слоя сети.

Таким образом, каждый скрытый нейрон описывается парой параметров (σ_j, \mathbf{c}_j) . Вектор χ кодируется в двойном наборе хромосом каждой особи.

Оценка приспособленности особи проводится в несколько этапов:

1. Для каждого \mathbf{c}_j , $j = \overline{1, N_m}$ в полном тренировочном наборе Ξ ищется ближайший во входном пространстве шаблон $\{\mathbf{z}_j, \mathbf{d}_j\}$:

$$\|\mathbf{c}_j - \mathbf{z}_j\| \leq \|\mathbf{c}_j - \mathbf{z}_i\|, \quad \mathbf{z}_i \in \Xi, \quad i = \overline{1, N} \quad (2.21)$$

2. По найденным шаблонам формируется максимальная тренировочная выборка $\Xi_m \left\{ \mathbf{c}_i, \mathbf{d}_i \right\}_{i=\overline{1, N_m}}$. Как видно, в отличие от классической

методики (2.6) центры активационных функций не соответствуют входным шаблонам, однако желаемая выходная реакция сети определяется по данным из ближайших шаблонов.

3. По вектору проектирования χ конструируется скрытый слой РБФ-сети (см. рис. 2.6). Для того чтобы определить, будет ли в сети присутствовать нейрон, представленный парой (σ_j, \mathbf{c}_j) , проверяются следующие два условия:

а) ширина окна активационной функции нейрона σ должна быть больше некоторого предельного значения $\varepsilon_\sigma > 0$:

$$\sigma_j > \varepsilon_\sigma, \quad j = \overline{1, N_m}, \quad (2.22)$$

если это условие нарушается, то нейрон не включается в скрытый слой;

б) интерполяционная матрица \mathcal{I} не должна быть особенной. Если при некотором $\varepsilon_\varphi > 0$ для двух нейронов оказывается, что:

$$\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{\sigma_i^2} < \varepsilon_\varphi, \quad \frac{\|\mathbf{c}_j - \mathbf{c}_i\|^2}{\sigma_j^2} < \varepsilon_\varphi, \quad i = \overline{1, N_m}, \quad i \neq j, \quad (2.23)$$

то один из нейронов не включается в скрытый слой. Если $\sigma_i > \sigma_j$, то игнорируется j -й нейрон, и наоборот.

Нейроны со слишком узкими окнами необходимо элиминировать по двум причинам. Во-первых, этот нейрон является лишним, так как его вклад в преобразование, осуществляемое сетью, может быть компенсирован соседними нейронами с более широкими окнами. Во-вторых, наличие таких нейронов делает сеть избыточной, что негативно сказывается на гладкости аппроксимационной поверхности и снижает возможность обобщения новых данных.

Второе условие предохраняет сеть от двух ситуаций. Первая, когда два нейрона имеют одинаковые координаты центров активационных функций; вторая, когда два нейрона обладают столь широкими окнами,

что их реакция в скрытом слое сети является одинаковой. В первой ситуации интерполяционная матрица становится особенной и решение (2.10) просто невозможно. Во второй ситуации в интерполяционной матрице образуются две практически одинаковые строки. Матрица становится близкой к особенной, вследствие чего решение (2.10) дает весовые коэффициенты, различающиеся между собой на 7—10 порядков. На практике это приводит к вычислительной неустойчивости РБФ-сети. Из-за ошибок округления реакция сети становится непредсказуемой, и такая сеть не может обобщать.

Значения ограничений ε_σ и ε_φ выбираются на практике и зависят от конкретной задачи. Для рассматриваемых в работе задач они принимаются равными $\varepsilon_\sigma=0,01$, $\varepsilon_\varphi=0,001$.

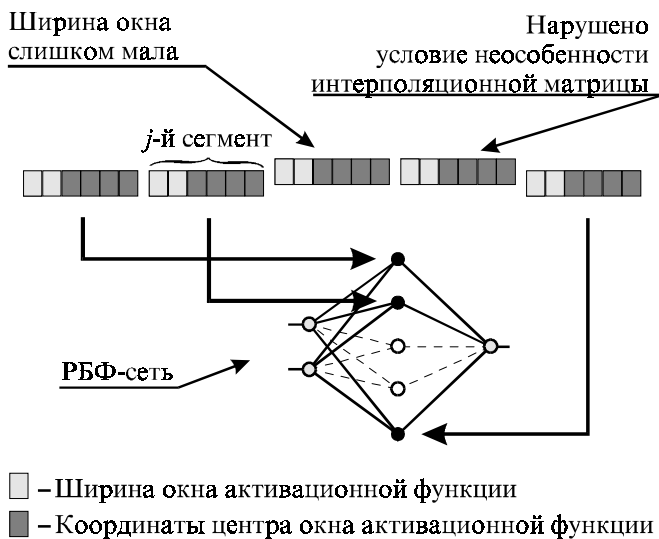


Рис. 2.6

Кодировка параметров скрытого слоя РБФ-сети в хромосоме

4. После того, как скрытый слой сформирован, определяется окончательная длина тренировочной выборки $N_r \leq N_m$. В результате ис-

ключения из Ξ_m шаблонов, соответствовавших не вошедшим в скрытый слой нейронам, формируется выборка Ξ_r , по которой проводится определение весов w_j , $j = \overline{1, N_r}$.

5. Приспособленность особи определяется, как ошибка реакции сети на всех шаблонах полного тренировочного набора Ξ , по следующей формуле:

$$E = \frac{1}{N} \sum_{i=1}^N \|\mathbf{d}_i - F(\mathbf{z}_i)\|, \quad (2.24)$$

где реакция РБФ-сети $F(\mathbf{z})$ вычисляется по (2.18).

Минимизация этой ошибки с помощью ГА, $f(\chi) = E$, приводит к синтезу сети, отвечающей условию (2.19). Алгоритм процедуры синтеза представлен на рис. 2.7.

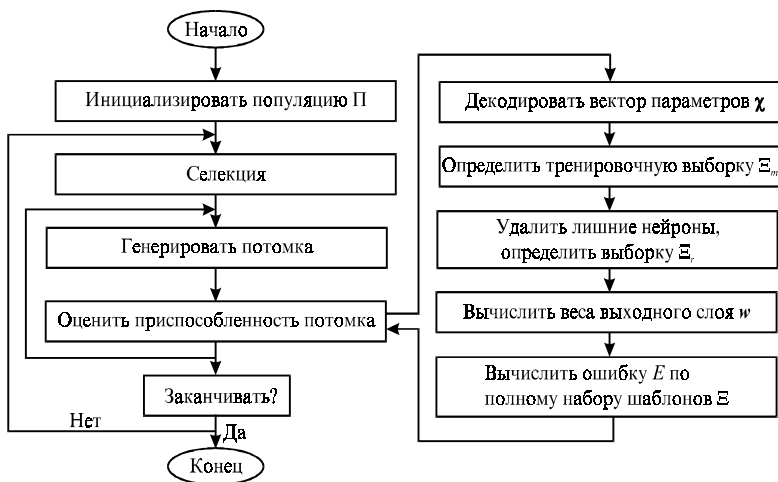


Рис. 2.7

Комбинированная процедура тренировки РБФ-сети

Необходимо отметить, что в ходе минимизации ошибки E ГА самостоятельно начинает генерировать координаты центров активационных функций нейронов скрытого слоя в точках, совпадающих с вход-

ными шаблонами. Если по методике, описанной в подразделе 2.1, центры принудительно размещались в шаблонах, то теперь они сами стремятся к ним. Это лишний раз подтверждает оптимальность аппроксимационной поверхности, рассчитываемой по (2.10).

Вероятно, на начальном этапе работы описанного алгоритма несовпадение центров и входных шаблонов вызывает некоторое ухудшение аппроксимационных свойств генерируемых сетей. Тем не менее, свобода, предоставляемая ГА в вопросе выбора координат центров, позволяет ему непрерывно перемещать поисковые точки в пространстве входов. В то время как жесткое связывание центров с шаблонами не только вызывает значительные алгоритмические трудности, но и, в конечном счете, сильно затрудняет поиск наилучшего решения.

Максимально возможный размер тренировочной выборки N_m определяется в зависимости от сложности рассматриваемой задачи, и на практике ограничен 100. Как показывает опыт, произвольно выбранное значение N_m всегда завышено, что приводит к избыточности размера конечной сети. С помощью ГА можно решить также и задачу синтеза РБФ-сети минимальной структуры, если добавить к ошибке E штраф за размер скрытого слоя сети:

$$E_m = E + \theta N_r, \quad (2.25)$$

где θ — вес штрафа.

Решая задачу (2.18, 2.19) с целевой функцией (2.25) ГА определяет наиболее репрезентативную выборку шаблонов наименьшего размера. Синтезируемая по ней РБФ-сеть имеет минимальную структуру, помнит все тренировочные шаблоны и имеет гарантируемый уровень обобщения.

2.4. Синтез нейроэмулятора динамического объекта

Используя описанную в предыдущем подразделе методику, построим НЭ тестового динамического объекта, описываемого дифференциальным уравнением:

$$\begin{aligned} T^2 x'' + 2T\zeta x' + x &= ku; \\ y &= x; \end{aligned} \quad (2.26)$$

где $T = 0,5$; $\zeta = 0,1$; $k = 1$.

Это — осциллятор, параметры которого выбраны так, чтобы тестовый объект обладал явно выраженными колебательными динамическими свойствами.

Вообще, дифференциальное уравнение вида (2.26) широко используется в теории автоматического управления для упрощенного линейного описания различных динамических объектов управления. Именно это определяет его выбор в качестве тестового объекта для построения нейросетевой идентификационной модели. Хотя линейный осциллятор и не может продемонстрировать все возможности ИНС, с его помощью можно выработать методику синтеза НЭ, общую для широкого класса объектов, обладающих колебательными переходными характеристиками [68].

Сначала необходимо синтезировать по (2.26) тренировочный набор шаблонов так, как если бы он формировался в ходе экспериментов с реальным объектом [69]. Для этого будем подавать на вход тестового объекта сигналы вида:

$$u_{ij} = A_i \cos(\omega_j t). \quad (2.27)$$

Пусть рабочий диапазон входных воздействий ограничен:

$$u \in [-1, 1], \quad (2.28)$$

тогда выберем следующий набор параметров тестового сигнала управления (2.27):

$$A_i = -1 + 0,5i, \quad i = \overline{0,4}$$

$$\omega_j = 0,001 \cdot 2^j, \quad j = \overline{0,14} \quad (2.29)$$

При $j=0$ ($\omega=0$) u_{ij} превращается в скачкообразную функцию $A1(t)$, и соответствующая ей выходная реакция объекта $y=x$ позволяет судить о статических характеристиках объекта. При $j>0$ входной сигнал является гармоническим, а реакция объекта показывает его динамические свойства. Так как для $j=14$ ($\omega=16,38 \text{ с}^{-1}$) амплитуда выходного сигнала объекта y уже в 100 раз меньше амплитуды входного, то можно игнорировать воздействия с большей частотой, из-за их низкой информативности. Равномерная распределенность амплитуд 75-и тестовых входных сигналов по допустимому диапазону позволяет получить достаточно полное описание статических и частотных характеристик динамического объекта.

Выберем шаг предсказания НЭ:

$$\Delta=0,05 \text{ с}, \quad (2.30)$$

где $t_{k+1}=t_k+\Delta$.

Полученные при подаче на вход объекта тестовых сигналов переходные процессы дискретизируем по времени с таким же шагом. При длительности сигнала управления u 20 с, общее количество экспериментальных данных составляет 30000.

Для идентификации (2.26) положим $l_1=3$, $l_2=0$, и запишем уравнение нейросетевой идентификационной модели (1.22) в виде:

$$\hat{y}(k+1) = NE(\hat{y}(k), \hat{y}(k-1), \hat{y}(k-2), \hat{y}(k-3), u(k)), \quad (2.31)$$

где NE — проводимое сетью преобразование вход-выход.

Здесь используется « NE » вместо « NN », чтобы подчеркнуть применение ИНС в качестве НЭ. На рис. 2.8 представлена соответствующая (2.31) схема НЭ тестового динамического объекта на базе 5- h -1 РБФ-сети.

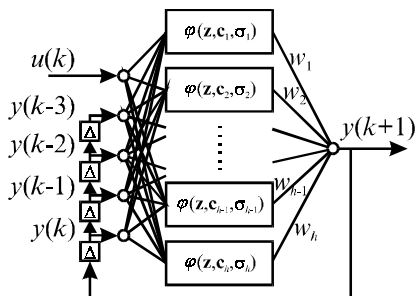


Рис. 2.8

Структура нейроммулятора на базе 5-h-1 РБФ-сети

Необходимо заметить, что нейросетевая идентификационная модель (2.31) заведомо выбрана избыточной по сравнению с конечноразностным представлением уравнения (2.26). Тем самым задача приближена к реальным условиям, так как обычно порядок идентифицируемой динамической системы завышается при оценивании.

Из (2.31) видна структура тренировочного шаблона $\{z, d\}$ для ИНС:

$$z = (y(k), y(k-1), y(k-2), y(k-3), u(k)), \quad d = y(k+1) \quad (2.32)$$

В рамках этой структуры по полученным экспериментальным данным формируется тренировочный набор шаблонов Ξ . Однако прежде чем использоваться для обучения сети он должен быть дополнительно обработан.

Дело в том, что среди 30000 шаблонов в наборе Ξ есть одинаковые. Так например, одинаковы все шаблоны, сформированные по реакции объекта на сигнал $u=0$ (он соответствует $A_2=0$). При подаче на вход гармонического сигнала одинаковыми являются шаблоны, полученные по отстоящим друг от друга на период точкам переходного процесса в установившемся режиме. Наличие в наборе таких шаблонов затрудняет работу процедуры обучения.

Чтобы исключить из набора одинаковые или близкие друг к другу шаблоны, вводится процедура сжатия. Суть ее состоит в устранении шаблонов $\{\mathbf{z}, d\}_i$, $i \in [1, N]$, не отвечающих следующему условию:

$$\|\mathbf{z}_i - \mathbf{z}_j\| \geq \varepsilon_c; \quad j = \overline{1, N}; \quad i \neq j, \quad (2.33)$$

где \mathbf{z}_i — входная часть тренировочного шаблона $\{\mathbf{z}, d\}_i$,

N — размер тренировочного набора Ξ ,

$\varepsilon_c > 0$ — радиус сжатия, определяющий минимально допустимое расстояние между двумя шаблонами.

Норма $\|\cdot\|$ вычисляется следующим образом:

$$\|\mathbf{z}\| = \sum_{k=1}^g (\beta_k z_k)^2, \quad (2.34)$$

где β_k — вес, нормирующий координаты вектора \mathbf{z} к единичному значению:

$$\beta_k = (\sup z_k - \inf z_k)^{-1}. \quad (2.35)$$

Для данной задачи веса выбираются равными: $\beta_k = 0,1$ $k = \overline{1, 4}$, $\beta_5 = 0,5$.

Хотя выполнение условия (2.23) и гарантирует защиту интерполяционной матрицы от вырождения в случае одинаковых или близко расположенных центров активационных функций скрытых нейронов, исключение совпадающих шаблонов значительно облегчает работу метода обучения РБФ-сети на базе ГА.

Вообще, необходимо отметить, что нейросетевые технологии требуют использования значительных вычислительных ресурсов. Как и любая оптимизационная задача, обучение ИНС связано с большим количеством итеративно повторяющихся вычислений. Для того, чтобы процесс обучения шел быстрее, тренировочный набор шаблонов должен быть как можно меньшим, но в тоже время репрезентативным. Поэтому

процедура сжатия является необходимой, особенно при работе с экспериментальными данными.

Для данной задачи выберем $\varepsilon_c=0,05$. Важно, чтобы величина ε_c не была слишком большой, иначе набор шаблонов потеряет репрезентативность. В результате сжатия размер Ξ уменьшился до 1000, что позволяет достаточно быстро обрабатывать тренировочный набор даже на персональном компьютере.

Как описано в подразделе 2.3, представим параметры сети в виде вектора проектирования:

$$\chi = (\sigma_1, c_{1,1}, c_{2,1}, c_{3,1}, c_{4,1}, c_{5,1}, \dots; \sigma_{50}, c_{1,50}, c_{2,50}, c_{3,50}, c_{4,50}, c_{5,50}), \quad (2.36)$$

здесь $\mathbf{c}_j = (c_{1j}, c_{2j}, c_{3j}, c_{4j}, c_{5j})$, и $N_m=50$. Область поиска ГА X_r определим следующим образом:

$$\sigma_j \in [0, 100]; c_{ij} \in [-4,8; 4,8], i=1,2,3,4; c_{5j} \in [-1, 1], j = \overline{1,50} \quad (2.37)$$

ОДЗ для координат центров активационных функций выбирается такой, чтобы она охватывала все имеющиеся входные шаблоны. Нижняя граница допустимой ширины окон принимается равной нулю. Максимальное значение окна, как показывает практика, должно быть выбрано таким, чтобы один нейрон мог «перекрыть» всю ОДЗ координат центров, но не более:

$$\sigma_j = k \sqrt{\sum_{i=1}^g (\sup c_{ij} - \inf c_{ij})^2}, j = \overline{1, N_m}. \quad (2.38)$$

Величина коэффициента k лежит в диапазоне $k \in [1, 20]$, так что значение активационной функции при максимальной ширине окна уменьшается в пределах области поиска на 0,05—10 %.

Вообще, границы поисковой области оказывают большое влияние на скорость и точность синтеза РБФ-сети с помощью ГА. Так например, если из-за слишком большой ширины окна нейтрализуется j -й

нейрон, то в векторе проектирования χ игнорируется вся группа (σ_j, c_j) . Следовательно, при любых изменениях параметров c_j с сохранением того же σ_j целевая функция (2.24) будет оставаться постоянной. Такая инвариантность негативно сказывается на эффективности поиска оптимальной конфигурации и настройки сети, поэтому важно сразу ограничить максимальную ширину окон активационных функций.

Выберем длину каждой хромосомы $\lambda_i=10$, $i = \overline{1,300}$. Сгенерировав популяцию из 100 особей, выполним обучение НЭ с помощью ГА.

В табл. 2.1 приведены параметры синтезированной РБФ-сети. НЭ содержит в скрытом слое 15 нейронов из 50 возможных, при этом сеть помнит более 30000 шаблонов. Средняя ошибка на одном шаблоне составляет 0,005 (0,5 % от желаемого значения). Так как синтез НЭ проводился с использованием штрафа за размер скрытого слоя (2.25), то можно считать, что НЭ обладает минимальной структурой, обеспечивающей указанную точность.

Таблица 2.1

Параметры нейроэмулятора тестового динамического объекта

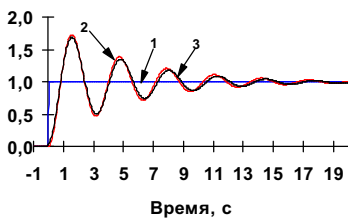
Номер нейрона	σ_i	c_{1i}	c_{2i}	c_{3i}	c_{4i}	c_{5i}	w_i
1	10,7038	1,7314	1,7877	1,8158	1,8440	-0,4858	0,8999
2	21,1730	4,2088	4,0211	3,7865	3,5238	0,2493	5,3091
3	90,1466	2,6041	2,5384	2,4446	2,3320	0,1105	- 14825,2096
4	18,6217	-1,4405	-1,5437	-1,6563	-1,7220	0,3607	1,9287
5	86,0117	0,1267	0,1736	0,2205	0,2581	-0,6246	20775,0037
6	78,8856	2,7824	-3,1390	3,0358	2,4540	0,0381	-508,6417
7	91,7302	-0,7836	-4,8000	-0,3144	-0,0798	-0,9316	1147,3173
8	10,1760	1,2340	1,5531	1,8440	2,1161	-0,9863	-1,3651

9	87,5073	-2,5384	-2,4446	-2,3320	-2,1912	-0,1887	-
							22575,9268
10	96,6569	-0,7836	-0,9713	-1,1402	-1,2997	0,9413	15229,9902
11	15,8065	-3,6551	-3,5801	-3,4768	-3,3267	-0,0068	3,2982
12	13,4311	4,4059	4,2370	4,0587	3,7677	0,9941	3,6839
13	52,3167	-4,2182	-4,0680	-3,8991	-3,6457	-0,1222	661,1589
14	10,6158	-1,5718	-1,5812	-1,5718	-1,5531	-0,8123	-1,5523
15	10,0880	-2,3226	-2,3320	-0,0047	-2,3226	0,0909	0,9065

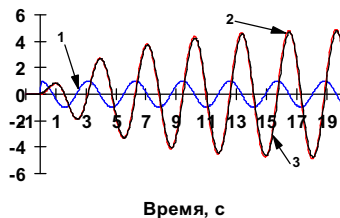
На рис. 2.9 приведены переходные процессы в тестовом динамическом объекте и его НЭ при различных входных сигналах. Как видно, синтезированный НЭ с приемлемой точностью описывает реакцию осциллятора при скачкообразном и гармоническом входе.

Разработанная методика синтеза РБФ-сети опирается на анализ ее способности предсказывать поведение динамического объекта только на один шаг вперед. То есть НЭ обучается как одношаговый предиктор. Однако, благодаря тому, что эта методика обеспечивает синтез сети с гарантированным высоким уровнем обобщения, полученный НЭ устойчиво работает и как краткосрочный предиктор. Более того, несмотря на то, что в тренировочном наборе шаблонов присутствовали данные о переходных процессах только в течение 20 с, реальная глубина предсказания НЭ с сохранением достоверности превышает $T_m=100$ с.

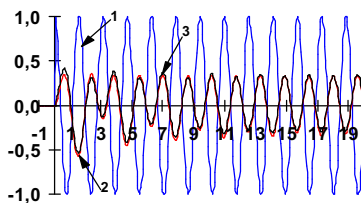
Благодаря высокому уровню обобщения синтезированной РБФ-сети, НЭ работает корректно не только для входных воздействий, использовавшихся при создании тренировочного набора шаблонов, но и во всем рабочем диапазоне $u \in [-1,1]$. Дискретность предсказания, осуществляемого НЭ, ограничивает допустимую частоту входного воздействия, поэтому синтезированный НЭ корректно работает при $\omega \leq 20$ с⁻¹.



а)



б)



Время, с

в)

Рис. 2.9

Переходные процессы в тестовом динамическом объекте и его нейроэмуляторе.

Кривая 1 — тестовое входное воздействие $u=\cos(\omega t)$;

кривая 2 — реакция динамического объекта;

кривая 3 — реакция нейроэмулятора.

а) $\omega=0$; б) $\omega=2$ и в) $\omega=4$

Синтез РБФ-сети потребовал около 10000 итераций ГА, при этом было проведено 50000 вычислений целевой функции (2.25). Время счета на компьютере класса Pentium-100 составило около 3 часов [70].

2.5. Идентификация динамического объекта по зашумленным данным

Как было показано в подразделе 2.4 предложенная методика обучения РБФ-сетей обеспечивает быстрое и эффективное решение тестовой задачи. Однако в случае реальных задач, характеризующихся наличием шумов в данных вход-выход, возникают особенности, которые могут привести к тому, что обученный НЭ не будет способен давать краткосрочный прогноз из-за низкого уровня обобщения в нем.

Прямое контролируемое обучение ИНС может рассматриваться как задача восстановления многомерного преобразования по имеющимся данным вход-выход. В этом смысле синтез НЭ является обратной задачей. Причем если он проводится по данным, полученным с реальных объектов, то, согласно теории регуляризации Тихонова, обучение является плохо обусловленной задачей. Это определяется двумя факторами. Во-первых, наличие шума в тренировочных данных нарушает условие однозначности преобразования, то есть двум разным входным векторам может соответствовать один выходной. Во-вторых, нарушается условие непрерывности, так как небольшое изменение входного вектора может сопровождаться большим изменением выходного. При этом тренировочные данные могут находиться вне области допустимых значений и области определения.

Решение задачи идентификации реальных динамических объектов всегда проходит в несколько попыток [71, 72]. Ограниченность вычислительных ресурсов и времени естественно приводят к тому, что процесс поиска структуры и параметров идентификационной модели идет от простого к сложному. То есть вначале выбирается простая модель процесса, а затем она усложняется, пока не будет достигнута требуемая точность.

При обучении РБФ-сети главным параметром, влияющим на ее информационную емкость и качество аппроксимации, является размер скрытого слоя. Вследствие этого при получении неработоспособного НЭ естественным действием выглядит увеличение максимально возможного размера скрытого слоя РБФ-сети N_m . Однако это может не только не улучшить гладкость и точность аппроксимации, но и привести к обратному результату. Если число используемых тренировочных шаблонов N_r (то есть размер скрытого слоя) значительно превышает число степеней свободы идентифицируемого динамического процесса, то задача определения выходных весов сети оказывается переопределенной [30]. Вследствие этого аппроксимационная поверхность, описываемая сетью, искажается, стремясь пройти через все используемые тренировочные шаблоны. Так как выходные шаблоны уже содержат шумы, то сеть теряет способность фильтровать их и, как следствие, обобщать новые.

Учитывая плохую обусловленность задачи обучения, наиболее критичными параметрами, определяющими успешность идентификации, являются глубины задержек обратных связей l_1, l_2 по выходу НЭ \hat{y} и по управляющему воздействию u , соответственно. Размер скрытого слоя влияет на точность идентификации, но не определяет ее возможность.

Исходя из этого, при идентификации нового объекта, необходимо сначала выбрать $N_m=10-30$ и увеличивать l_1, l_2 , пока динамические характеристики нейросетевой модели не приблизятся к характеристикам идентифицируемого объекта. Затем должен быть выбран обеспечивающий требуемую точность модели максимальный возможный размер скрытого слоя N_m .

Проиллюстрируем описанную методику на примере задачи синтеза НЭ того же тестового динамического объекта (2.26) [73]. Однако в

этот раз, будем учитывать, что на практике при измерении выходных сигналов объекта их значения искажаются под действием внешних помех. Для этого при формировании тренировочного шаблона $\{z, d\}$ будем использовать не истинные значения сигнала y , а их измерения:

$$y = x + \theta. \quad (2.39)$$

Здесь θ — ошибка измерения, являющаяся случайной величиной. Ее математическое ожидание примем равным нулю, а дисперсию равной 0,05.

Выберем шаг предсказания НЭ равным $\Delta = 0,05$ с и соберем, используя тестовые сигналы (2.27)—(2.29), экспериментальные данные о поведении динамического объекта.

Вначале, выполняя те же шаги, что и в предыдущем подразделе, попытаемся идентифицировать объект в рамках модели (2.31), то есть при $l_1=3, l_2=0$. На рис. 2.10,а представлены переходные функции синтезированного на базе 5-10-1 РБФ-сети НЭ и самого объекта. Как видно, не смотря на то, что порядок нейросетевой идентификационной модели (2.31) выше порядка динамического объекта, точность полученной модели низка.

Тем не менее, если использовать полученный эмулятор как одношаговый предиктор по формуле:

$$\hat{y}(k+1) = NE(y(k), y(k-1), y(k-2), y(k-3), u(k)), \quad (2.40)$$

то ошибка предсказания составит менее 0,1 % (см. рис. 2.10,б). Это доказывает, что обучение выполнено полностью, и ИНС достаточно точно запомнила тренировочные шаблоны [74]. Следовательно, нет необходимости далее увеличивать возможный размер скрытого слоя. Наличие шумов в тренировочных шаблонах снижает точность модели, уменьшая ее глубину предсказания, и чтобы решить эту проблему необходимо повысить порядок самой идентификационной модели.

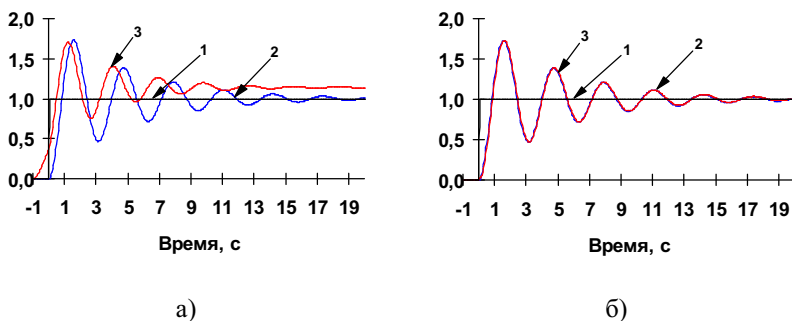


Рис. 2.10

Моделирование переходной функции тестового динамического объекта при недостаточной сложности нейросетевой идентификационной модели.

Кривая 1 — тестовое входное воздействие $u=1(t)$;

кривая 2 — реакция динамического объекта;

кривая 3 — реакция нейроэмулятора.

а) — работа нейроэмулятора в режиме краткосрочного предиктора;

б) — в режиме одношагового предиктора

Результаты экспериментов показывают, что точность идентификации, главным образом, зависит от глубины задержки обратной связи по выходу НЭ l_1 . Так при $l_1=5$ точность идентификационной модели выше, чем в предыдущем случае, однако все еще недостаточна. Поэтому положим $l_1=7$, $l_2=0$.

Для такой нейросетевой идентификационной модели тренировочный шаблон $\{z, d\}$ имеет структуру:

$$z = (y(k), y(k-1), \dots, y(k-7), u(k)), \quad d = y(k+1) \quad (2.41)$$

Сжимая тренировочный набор шаблонов Ξ по описанной в подразделе 2.4 методике с радиусом сжатия $\varepsilon_c=0,05$, получим окончательный тренировочный набор шаблонов размером $N=1000$.

Как было установлено ранее, 10-и нейронов скрытого слоя вполне достаточно, чтобы запомнить все тренировочные шаблоны, поэтому определим максимально допустимый размер скрытого слоя НЭ равным $N_m=10$. Результаты экспериментов показывают, что если N_m выбрать большим, то задача обучения НЭ окажется переопределенной, и качество аппроксимации ухудшится. Таким образом, в идеальном случае незашумленных тренировочных данных величина максимального размера скрытого слоя ограничивается лишь требуемыми вычислительными затратами на моделирование сети и ее избыточность не влияет на конечный результат обучения НЭ. Если идентификация динамического объекта проводится по реальным данным, содержащим помехи, то N_m является одним из критических параметров и не может выбираться произвольно большим.

Синтезируемый НЭ имеет 9 входных нейронов, один выходной, и описывается вектором проектирования:

$$\chi = (\sigma_1, c_{1,1}, c_{2,1}, \dots, c_{9,1}; \dots; \sigma_{10}, c_{1,10}, c_{2,10}, \dots, c_{9,10}), \quad (2.42)$$

Область поиска ГА X_r определим следующим образом:

$$\sigma_j \in [0, 200]; c_{ij} \in [-4,9; 4,9], i = \overline{1,8}; c_{9,j} \in [-1, 1], j = \overline{1,10} \quad (2.43)$$

Длину каждой хромосомы, определим равной $\lambda_r=10$, $i = \overline{1,10}$.

Сгенерировав популяцию из 100 особей, выполним обучение НЭ. В табл. 2.2—2.3 приведены параметры синтезированной РБФ-сети.

Таблица 2.2

Параметры нейроэмулятора σ , $c_1—c_5$

Номер нейрона	σ_i	c_{1i}	c_{2i}	c_{3i}	c_{4i}	c_{5i}
1	57,3607	-0,6188	-0,2730	-0,1485	-0,0048	0,1197
2	10,5572	-0,2434	-0,5317	-0,5891	-0,6275	-0,6849
3	82,6197	1,0000	-1,3555	-1,7004	-2,0261	-2,3326

Продолжение табл. 2.2

4	125,8944	-0,9687	4,9000	-0,1485	-0,3018	-0,3018
5	36,5591	-0,8925	-0,9915	-0,6083	-0,6083	-0,1485
6	52,1603	-0,0655	-3,9133	-3,7217	-2,4955	-3,5876
7	152,6393	0,4076	3,6451	3,3768	3,0703	2,7446
8	170,6549	0,9531	0,4359	0,4646	0,5029	0,6083
9	92,4633	-0,1105	-4,4114	-4,2294	-4,0283	-3,7600
10	33,5875	0,5015	3,4630	3,1373	2,7637	2,3805

Таблица 2.3

Параметры нейроэмулятора $c_6—c_9, w$

Номер нейрона	c_{6i}	c_{7i}	c_{8i}	c_{9i}	w_i
1	0,2347	0,3688	0,5317	0,6083	-346,2565
2	-0,7328	-0,7807	-0,8286	-0,8765	0,0069
3	-2,7542	-2,8787	-3,0703	-3,3098	283,8242
4	-0,6083	-0,0048	-0,5413	-0,0048	100,0852
5	-0,1485	-0,0335	0,6083	4,7563	35,8334
6	-3,2236	-2,9745	-2,6967	-2,3997	-22,5358
7	2,3805	1,9974	1,5854	1,1831	2553,0869
8	0,6083	0,6275	0,6083	0,7616	-2803,9732
9	-3,4343	-3,2140	-2,7733	-2,3805	206,1569
10	1,9878	1,5184	1,0777	0,6083	4,2962

НЭ содержит в скрытом слое 10 нейронов из 10 возможных, при этом сеть помнит более 30000 шаблонов. Несмотря на возросшую сложность задачи, оптимальные значения настроечных параметров НЭ, как и в предыдущем примере, были получены примерно за 50000 вычислений целевой функции в течение 3 часов счета на персональном компьютере класса Pentium-100.

Как показывает рис. 2.11 синтезированный НЭ достаточно точно описывает идентифицируемый динамический объект и может использоваться как краткосрочный предиктор.

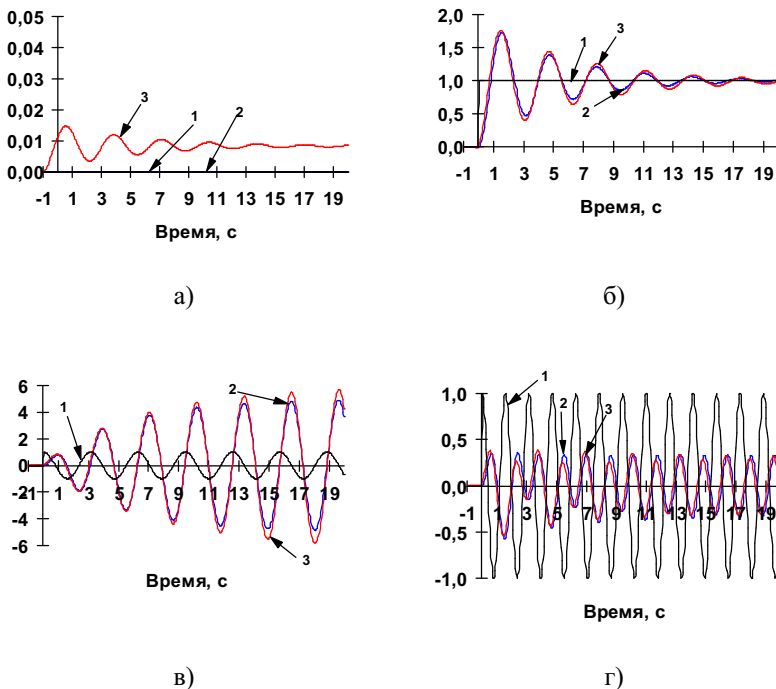


Рис. 2.11

Переходные процессы в тестовом динамическом объекте и его нейроэмуляторе, построенном по зашумленным данным.

Кривая 1 — тестовое входное воздействие $u=A\cos(\omega t)$;

кривая 2 — реакция динамического объекта;

кривая 3 — реакция нейроэмулятора;

а) — $A=0, \omega=0$; б) — $A=1, \omega=0$; в) — $A=1, \omega=2$ и г) $A=1, \omega=4$

Ошибка предсказания выходного сигнала объекта не превышает 5 % для скачкообразных входных воздействий лежащих в диапазоне

$u \in [-1,1]$ (см. рис. 2.11,а,б) и гармонических, имеющих частоту $\omega \leq 10 \text{ с}^{-1}$ (см. рис. 2.11,в,г). Глубина предсказания НЭ превышает 100 с.

По сравнению с НЭ, обучавшимся по незашумленным данным, данная нейросетевая идентификационная модель менее точна. НЭ обладает статической ошибкой, равной 0,008 (см. рис. 2.11,а), и фазовым сдвигом, не превышающим 5° . Тем не менее, полученный результат следует считать положительным, так как достигнутая точность модели вполне приемлема для проведения исследования свойств динамического объекта и позволяет синтезировать по ней систему управления.

В целом, приведенный пример доказывает возможность идентификации динамических объектов с помощью ИНС по реальным данным, содержащим шумы. При этом наиболее критическим параметром является сложность самой нейросетевой идентификационной модели, а размер скрытого слоя выступает вторичным фактором.

2.6. Выводы

Разработана методика эволюционного синтеза РБФ-сетей, представляющая собой симбиоз метода глобальной оптимизации параметров скрытого слоя и локального метода настройки входных весов. На примере задачи построения НЭ осциллятора сформулированы правила выбора параметров нейросетевых идентификационных моделей.

Так как тестовый динамический объект обладает невысоким порядком и является линейным, его дифференциальная модель структурно более проста, чем нейросетевая. Однако нейросетевые технологии и не направлены на вытеснение классических подходов. Модели на базе ИНС выглядят внешне более громоздко, чем интегро-дифференциальные и имеют на один—два порядка большее количество настроечных параметров. Тем не менее, являясь существенно нелиней-

ными по своей природе, ИНС позволяют идентифицировать как линейные, так и нелинейные динамические объекты.

Пример решения задачи идентификации динамического объекта по зашумленным данным показывает, что для синтеза НЭ не нужны дополнительный статистический анализ входных и выходных сигналов объекта и исследования взаимосвязей его параметров. Выбор типа нейросетевой модели сводится прежде всего к определению глубин задержек обратных связей по ее входу и выходу, а затем размера скрытого слоя РБФ-сети.

В сочетании с разработанными правилами обработки тренировочных шаблонов данных, выбора размеров поискового пространства, а также защиты интерполяционной матрицы от вырождения эволюционная методика позволяет синтезировать РБФ-сети с гарантированным уровнем обобщения. Это делает возможным использование ее для построения точных краткосрочных предикторов с большой глубиной предсказания.

ГЛАВА 3

РАЗРАБОТКА МЕТОДИКИ СИНТЕЗА НЕЙРОСЕТЕВЫХ СИСТЕМ УПРАВЛЕНИЯ

Процесс синтеза системы управления непосредственно связан с выбором закона управления. Большинство классических методов теории управления направлено на одновременное получение вида и параметров закона управления, а также соответствующих ему структуры контроллера и самой СУ. Синтез НСУ, как и любой другой адаптивной СУ, проходит другим образом [75]. Вначале выбирается общий вид закона управления, то есть структура НСУ и поступающие на вход НК сигналы. Затем в процессе обучения (адаптации) определяются параметры НК. Важно, что при этом не нужно каким-либо образом задавать извне вид зависимости сигнала управления от входных сигналов НК. Благодаря свойству универсальной аппроксимации, преобразование вход-выход, обеспечивающее требуемое качество управления, определяется системой самостоятельно в ходе обучения [76, 77].

Так как желаемая реакция НК изначально неизвестна, то его параметры могут быть определены только в ходе стимулируемого обучения по оценке качества работы всей НСУ в целом. Наиболее подходящими для этого вида обучения являются прямонаправленные сети типа многослойный перцептрон. Для задач управления более удобными являются нечетные активационные функции, поэтому далее под МСП будем подразумевать МСП с тангенциальными активационными функциями (1.10) скрытых нейронов.

Рассмотрим задачу, решаемую тангенциальным нейроном скрытого слоя МСП. Полагая в (1.4) $s=0$, мы получим уравнение гиперплоскости в пространстве входов нейрона. На рис. 3.1 представлен случай

нейрона с двумя входами. Здесь прямая $w_0 + w_1z_1 + w_2z_2 = 0$ делит входное пространство на две части. В одной из них активационная функция нейрона $th(s)$ положительна, в другой — отрицательна.

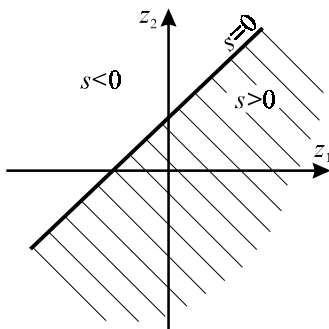


Рис. 3.1

Область определения тангенциальной активационной функции

Таким образом, каждый нейрон скрытого слоя классифицирует входные сигналы по двум классам. Однако в отличие от чисто классификационных задач, для нейроуправления важен не только сам факт изменения знака выходного сигнала нейрона, но и закон по которому этот сигнал меняется. Именно взвешенная сумма нелинейностей нейронов скрытого слоя определяет преобразование, осуществляемое сетью. При этом важным параметром является ширина активного диапазона изменения входных сигналов, в котором активационная функция еще не достигает насыщения.

В отличие от задачи обучения нейроэмулятора, использующей уже имеющиеся данные об объекте, синтез НК непосредственно связан с активным воздействием на объект. В связи с этим особое значение приобретает тип и количество тренировочных входных сигналов системы u_r (уставок), используемых при обучении сети.

Например, чтобы получить НК, корректно работающий в некотором непрерывном диапазоне амплитуд входных сигналов, необходимо,

чтобы в наборе тренировочных сигналов $\Sigma\{u_r^i\}_{i=1, M}$, где M — размер набора, присутствовали сигналы различной амплитуды из рассматриваемого диапазона. При этом их количество и распределение должно быть таким, чтобы сеть могла корректно обобщить (интерполировать) полученные знания на весь диапазон.

Для получения требуемых статических характеристик в тренировочный набор Σ должны входить ступенчатые сигналы, позволяющие точно оценить статическую ошибку НСУ. С помощью тренировочных гармонических сигналов можно задать системе требуемые динамические свойства. При этом распределение частот сигналов по рабочему диапазону также определяет качество проводимого НК обобщения.

Правильный выбор тренировочных сигналов полностью определяет качество НСУ и эффективность ее синтеза. Чрезмерно большой набор тренировочных сигналов увеличивает время синтеза НК, недостаточно полный набор делает невозможным использование НСУ в непрерывном спектре сигналов.

3.1. Использование генетического алгоритма для обучения нейроконтроллеров

Рассмотрим задачу синтеза НСУ типа «один вход — один выход» с НК на базе трехслойного МСП структуры $g-h-1$ [52]:

Для данного набора тренировочных сигналов $\Sigma\{u_r^i\}_{i=1, M}$, необходимо найти такие значения синаптических параметров \mathbf{w} , что реализуемый нейроконтроллером закон управления:

$$u = NC(\mathbf{w}, u_r), \quad (3.1)$$

обеспечивает минимальное значение функционала I :

$$I(y(u)) \rightarrow \min_{\mathbf{w}}. \quad (3.2)$$

Здесь вектор

$$\mathbf{w} \left(w_{0,1}, w_{1,1}, \dots, w_{g,1}; w_{0,2}, w_{1,2}, \dots, w_{g,2}; \dots; w_{0,h}, w_{1,h}, \dots, w_{g,h}; w_0, w_1, \dots, w_h \right) \quad (3.3)$$

определяет параметры нейронов скрытого w_{ij} , $i = \overline{0, g}$, $j = \overline{1, h}$ и выходного w_i , $i = \overline{0, h}$ слоев. Закон управления (3.1) принадлежит к типу (1.25), а функционал I вычисляется в зависимости от типа НСУ.

Для НСУ с эталонной моделью:

$$I(y) = \sum_{i=1}^M \sum_{k=0}^{k_\tau} \|\tilde{y}_i(k) - y_i(k)\|, \quad (3.4)$$

где \tilde{y}_i и y_i — реакция эталонной модели и объекта управления на тренировочное входное воздействие u_r^i , соответственно,

τ — длительность тренировочного сигнала $t \in [0, \tau]$, и $k_\tau \Rightarrow \tau = k_\tau \Delta$.

Здесь Δ — шаг управления, $t_{k+1} = t_k + \Delta$.

Для экстремальных следящих НСУ:

$$I(y) = \sum_{i=1}^M \sum_{k=0}^{k_\tau} \|y_i(k) - u_r^i(k)\| \quad (3.5)$$

Для решения поставленной задачи с помощью ГА выберем \mathbf{w} в качестве вектора проектирования:

$$\chi = \mathbf{w}, \quad (3.6)$$

а функционал I в качестве целевой функции:

$$f(\chi) = I. \quad (3.7)$$

На рис. 3.2 представлена соответствующая схема ГА.

Представленный метод обучения НК классифицируется как задержанное стимулируемое обучение. В принципе, с помощью ГА можно реализовать и методы непосредственного стимулируемого обучения, изменив соответствующим образом функционал (3.2) и выполняя поиск

оптимального вектора решения \mathbf{w} на каждом шаге работы НСУ. Однако сходимость таких методов уже не будет гарантированной из-за влияния многих дополнительных факторов, а значит будут потеряны все преимущества ГА перед локальными методами оптимизации. Поэтому на базе ГА традиционно строятся методы задержанного обучения. Эта особенность в большинстве случаев не позволяет эффективно использовать ГА для адаптивного управления в реальном масштабе времени. Однако в случае объектов управления с большими постоянными времени ГА может применяться и для синтеза контроллеров в реальном времени [78].

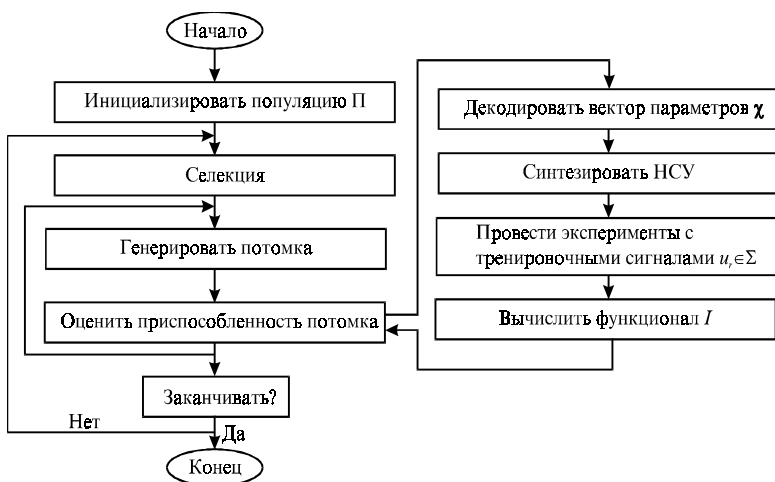


Рис. 3.2

Использование генетического алгоритма для обучения нейроконтроллера

Так как ГА работает в ограниченной области поиска, то для решения задачи (3.1, 3.2) необходимо определить верхнюю и нижнюю границы изменения параметров \mathbf{w} . Как показывает практика, выбор границ оказывает решающее влияние на эффективность работы ГА. Это касается, прежде всего, параметров скрытых нейронов. Хотя положение гиперплоскости, делящей входное пространство на два класса, опреде-

ляется только соотношением синаптических параметров, вид реализуемой нейроном нелинейности зависит от их абсолютных значений.

Тангенциальная активационная функция асимптотически стремится к своим предельным значениям при увеличении абсолютного значения ее аргумента. Если область поиска слишком широка, то для вариантов решения, обладающих большими значениями весов и сдвигов, осуществляемое сетью преобразование вход-выход становится нечувствительным к их изменению. Активационные функции скрытых нейронов таких сетей оказываются в «насыщенном» состоянии, и даже значительные вариации вектора проектирования не приводят к изменению целевой функции. Соответствующая этим решениям часть популяции перестает направлять поиск оптимального решения, и скорость обучения критически снижается.

Одним из способов упрощения процесса выбора области поиска может служить введение в вектор проектирования в явном виде параметра активационной функции a , определяющего ее наклон:

$$\varphi(s) = th(as). \quad (3.8)$$

В этом случае область поиска для синаптических параметров нейронов скрытого слоя можно выбрать равной $[-1,1]$. Учитывая максимальные и минимальные значения каждого входного сигнала сети, при условии, что ненасыщенный участок активационной функции $th(s)$ лежит в пределах $s \in [-3,3]$ (см. рис. 3.3), можно определить верхнюю границу изменения параметра a . Например, для рассматриваемых в работе задач оптимальное значение a искалось в области $[0,10]$.

В такой постановке задачи обучения вектор проектирования приобретает вид

$$\chi(a_1, w_{0,1}, w_{1,1}, \dots, w_{g,1}; \dots; a_h, w_{0,h}, w_{1,h}, \dots, w_{g,h}; w_0, w_1, \dots, w_h). \quad (3.9)$$

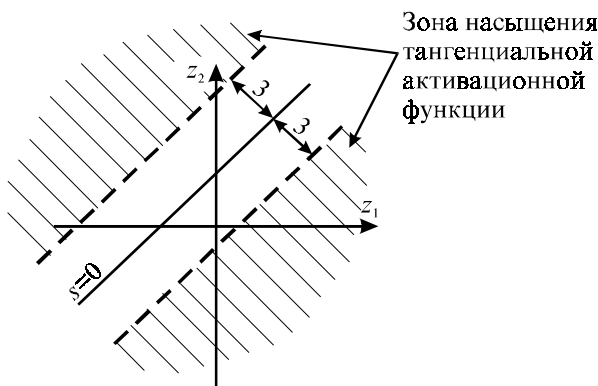


Рис. 3.3

Зона ненасыщенной активационной функции

Как показывает опыт, введение параметра a , не только облегчает выбор поискового пространства, но и способствует большей формализации задачи синтеза НСУ. Это делает предлагаемую методику более универсальной. Кроме того, явное присутствие параметра a в векторе проектирования разделяет для ГА процессы поиска оптимальной классификации входных сигналов сети и формы ее нелинейного преобразования. В результате этого, несмотря на возрастание размерности поисковой задачи, скорость обучения НК увеличивается.

Описанная методика синтеза НСУ с помощью ГА может использоваться и для многоканальных систем типа «несколько входов — несколько выходов». Для этого необходимо только соответствующим образом изменить вид функционала I .

3.2. Синтез нейросетевой системы управления с ПИД-нейроконтроллером

Рассмотрим задачу синтеза НСУ линейным динамическим объектом, использовавшимся в подразделе 2.4:

$$T^2 x'' + 2T\zeta x' + x = ku; \quad (3.10)$$

$$y = x;$$

где $T = 0,5$; $\zeta = 0,1$; $k = 1$.

На рис. 3.4 представлена переходная и амплитудно-частотная характеристики тестового объекта.

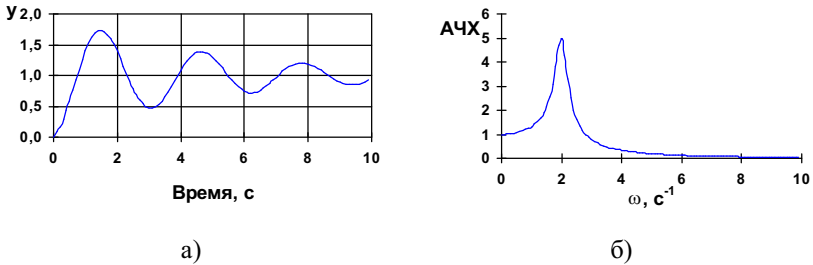


Рис. 3.4

Переходная и амплитудно-частотная характеристики тестового динамического объекта управления.

- а) — переходная характеристика объекта управления;
 б) — амплитудно-частотная характеристика

На сегодняшний день подавляющее большинство промышленных систем автоматического управления базируется на так называемых ПИД-контроллерах. Закон управления в этих системах представляет собой взвешенную сумму сигналов ошибки обратной связи по управляемой координате, ее интеграла и производной. В настоящее время существует множество методик определения весовых коэффициентов линейных и кусочно-линейных ПИД-контроллеров. В рамках такого закона управления разработаны методы синтеза адаптивных СУ и СУ с переменной структурой. Простота ПИД-контроллеров, с одной стороны, делает их привлекательными при создании новых автоматических СУ, но с другой стороны, она ограничивает их возможности. Использование

ИНС позволяет расширить сферу применения СУ с пропорционально-интегро-дифференциальным законом управления [79].

При разработке новых методов управления важно, чтобы они могли быть интегрированы в существующие технические комплексы, улучшая качество их работы без существенных затрат на реконструкцию. Нейроуправление представляет собой новое высокотехнологическое направление, требующее использования мощной вычислительной техники. Тем не менее, на базе ИНС могут создаваться простые НК, легко интегрируемые в существующие технологические комплексы. Примером такой системы является НСУ на базе ПИД-нейроконтроллера (ПИД-НК) (см. рис. 3.5) [80].

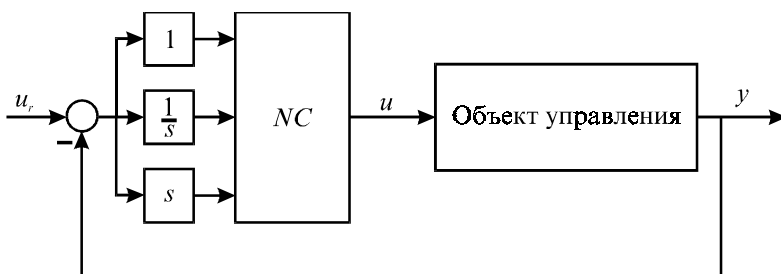


Рис. 3.5

Схема непрерывной нейросетевой системы управления

Это — непрерывная система. Сигнал управления u формируется в ней по следующей зависимости:

$$u = NC\left(e, \frac{de}{dt}, \int edt\right),$$

$$e = y - u_r. \quad (3.11)$$

Здесь введено обозначение « NC » вместо « NN », чтобы подчеркнуть применение ИНС в качестве НК.

ИНС, используемая в такой НСУ, выполняет только нелинейное преобразование и не содержит элементов задержки. Проектирование

НСУ сводится к определению размера скрытого слоя МСП и набора тренировочных сигналов. Хотя предложенная в подразделе 3.1 методика описана для случая дискретной НСУ, она может быть применена и для непрерывных систем. Решение с ее помощью задачи (3.1, 3.2) должно обеспечить требуемое качество управления.

Целью обучения ИНС является получение линейной СУ. При этом необходимо отметить, что хотя объект (3.10) также линейный, синтез ПИД-НК проводится без учета этого факта по общей для всего класса динамических объектов (линейных и нелинейных) методике. Реализуемый ПИД-НК закон управления (3.11) сам существенно нелинеен, поэтому линейность тестового объекта не ограничивает общности получаемых здесь результатов [81].

Для построения непрерывной НСУ сформируем следующий набор тренировочных сигналов Σ размером $M=5$:

$$\begin{aligned} u_r^i(t) &= A_i l(t), \quad t \in [0, \tau], \\ A_i &= -1,5 + 0,5i, \quad i = \overline{1,5}, \end{aligned} \quad (3.12)$$

где $l(t)$ — единичная функция Хевисайда.

Качество работы НСУ оценивается по функционалу вида (3.5). Модифицируя его для случая непрерывной СУ, получим:

$$I(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{\tau} \int_0^{\tau} \left(\frac{y_i(t) - u_r^i(t)}{|A_i| + \varepsilon} \right)^2 dt, \quad (3.13)$$

где ε — малый параметр, вводимый для обеспечения непрерывности функционала I .

$$\varepsilon = 0,01 \quad (3.14)$$

Как видно, в (3.13) введено усреднение вычисляемой интегральной оценки качества по количеству и длительности тренировочных входных сигналов. Также все переходные процессы нормируются на амплитуду соответствующего сигнала задания u_r . Такой вид функцио-

нала позволяет получать эквивалентные оценки качества работы НСУ при экспериментировании с разными наборами тренировочных сигналов.

Результаты сравнительных исследований показывают, что длительность тренировочных сигналов, а значит и переходных процессов в СУ существенно влияет на качество получаемого решения. Так при малом τ обучение, прежде всего, направлено на то, чтобы НК обеспечивал быстрый перевод объекта в состояние $y \approx u_r$. Однако колебательность и статическая ошибка такой системы остаются недопустимо большими. Увеличивая τ , можно добиться того, что процесс адаптации НК (то есть его обучения) приводит к получению более гладкого переходного процесса и нулевой статической ошибки. Однако чем больше τ , тем дольше идет оценка качества управления для каждого варианта решения, а значит увеличивается общее время синтеза НСУ. В данном случае для объекта (3.10) в качестве компромиссной длительности тренировочных входных сигналов было выбрано $\tau=10$ с.

Для построения нейросетевой части ПИД-НК выберем МСП структуры 3-5-1. Как видно из (3.11), на вход МСП поступают сигнал ошибки, его производная и интеграл. Нейроны скрытого слоя с тангенциальной активационной функцией осуществляют нелинейное преобразование и передают свои сигналы на единственный линейный выходной нейрон. Выходом сети является сигнал управления u .

Согласно (3.9) опишем обучаемый МСП вектором проектирования χ :

$$\chi = (a_1, w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; a_5, w_{0,5}, w_{1,5}, w_{2,5}, w_{3,5}; w_0, w_1, w_2, w_3, w_4, w_5). \quad (3.15)$$

Длину каждой хромосомы, соответствующей одной координате χ примем равной 10 битам.

Для оценки приспособленности особи по функционалу (3.13) уравнение осциллятора (3.10) преобразуется в форму Коши. Полученная система дифференциальных уравнений интегрируется явным методом Адамса 4-го порядка с постоянным шагом 0,01 с [82].

Далее определим область поиска X:

$$a_j \in [0,10], w_{ij} \in [-1,1], i = \overline{0,3}, j = \overline{1,5},$$

$$w_i \in [-10,10], i = \overline{0,5}. \quad (3.16)$$

Диапазон поиска по параметрам w_i зависит от необходимого максимального значения сигнала управления. В данном случае коэффициент усиления объекта управления единичный. Амплитуда тренировочных сигналов также равна единице, поэтому можно с уверенностью предположить, что управление с амплитудой в десять раз большей входного сигнала сможет обеспечить требуемые переходные процессы.

Выберем размер популяции равным 100 и проведем настройку параметров МСП с помощью ГА. В табл. 3.1, 3.2 приведены параметры нейросетевой части обученного ПИД-НК.

Таблица 3.1

Параметры нейросетевой части ПИД-нейроконтроллера.

Скрытый слой

Номер нейрона	a_i	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	0.5142	-0.7771	-0.9570	-0.8280	-0.1065
2	0.5024	0.5464	-0.4409	0.7576	-0.7791
3	0.7781	-0.3470	0.8788	0.7928	-0.2766
4	1.9042	-0.6481	-0.0538	-0.0166	-0.0068
5	0.9756	-0.2903	-0.9902	-0.5210	-0.5015

Параметры нейросетевой части ПИД-нейроконтроллера.

Выходной нейрон

Параметр	Значение	Параметр	Значение
w_0	0.1378	w_3	0.9922
w_1	-0.4018	w_4	0.9335
w_2	0.1554	w_5	-2.6100

На рис. 3.6 приведены графики переходных процессов в синтезированной НСУ. НСУ ведет себя линейно по отношению к величине входного воздействия, при этом качество переходных процессов одинаково для всех амплитуд. Несмотря на то, что тренировочные сигналы лежали в диапазоне $[-1, 1]$, генерируемое управление корректно для скачкообразных входных воздействий в диапазоне $[-2, 2]$. Это говорит о высоком уровне обобщения в обученной ИНС.

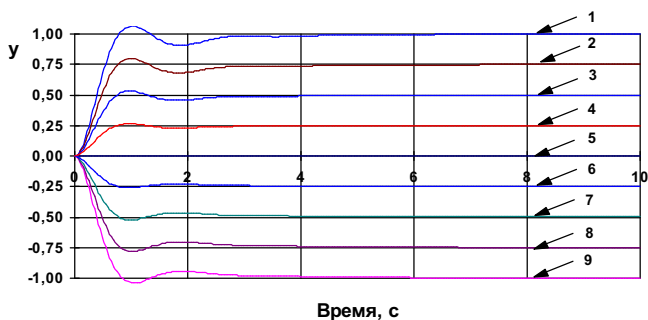


Рис. 3.6

Переходные процессы в НСУ с обученным ПИД-нейроконтроллером.

Входной сигнал $u_r = A1(t)$. Кривая 1 — $A=1$;

кривая 2 — $A=0,75$; кривая 3 — $A=0,5$;

кривая 4 — $A=0,25$; кривая 5 — $A=0$;

кривая 6 — $A=-0,25$; кривая 7 — $A=-0,5$;

кривая 8 — $A=-0,75$; кривая 9 — $A=-1$

При входных сигналах типа $A1(t)$, синтезированный ПИД-НК работает не хуже, чем линейный ПИД-контроллер при классической настройке на модульный оптимум: перерегулирование не превышает 5 %, статическая ошибка равна нулю, время регулирования мене 3 с, количество колебаний переходного процесса 1,5—2.

Тем не менее, анализ поведения НСУ при гармонических входных сигналах типа $A\sin(\omega t)$ показал ее неудовлетворительные частотные характеристики. На рис. 3.7 представлена амплитудно-частотная характеристика (АЧХ) НСУ. Положительным фактом является то, что ПИД-НК удалось подавить резонансные явления в объекте для $\omega \approx 2 \text{ с}^{-1}$. Однако переходные процессы в НСУ для гармонических входных сигналов при $\omega > 4 \text{ с}^{-1}$ (см. зона II рис. 3.7) показывают, что его работа приводит к появлению в выходном сигнале дополнительных гармоник. Кроме того, с ростом частоты входного воздействия быстро увеличивается фазовый сдвиг выходного сигнала объекта управления.

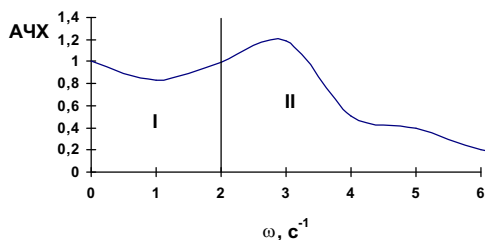


Рис. 3.7

Амплитудно-частотная характеристика НСУ.

I — область частот входных сигналов, отслеживаемых без искажений;

II — область частот входных сигналов, плохо отслеживаемых НСУ

Такие частотные характеристики синтезированной НСУ, объясняются составом использовавшегося тренировочного набора Σ . Полученные результаты показывают, что ПИД-НК не может обучаться толь-

ко по реакции НСУ на ступенчатые входные воздействия. Для синтеза НСУ с заданными амплитудно-фазо-частотными характеристиками (АФЧХ) в тренировочный набор необходимо включить также гармонические сигналы.

Обучение нейросетевой части ПИД-НК потребовало около 20000 вычислений функционала (3.13). При этом было проинтегрировано более 100000 переходных процессов. Общее время синтеза НСУ заняло около 1 часа вычислений на компьютере класса Pentium-100.

3.3. Нейросетевая система управления с заданными частотными характеристиками

Используя предложенную в подразделе 3.2 схему ПИД-НК (3.11), построим НСУ тестовым динамическим объектом (3.10), способную отслеживать с минимальным фазовым сдвигом не только скачкообразные, но и гармонические сигналы частотой $\omega \leq 3,5 \text{ с}^{-1}$ [83].

Для определенности потребуем, чтобы входные сигналы более высоких частот $\omega > 3,5 \text{ с}^{-1}$ отфильтровывались, то есть коэффициент усиления НСУ для таких сигналов был малым. На рис. 3.8 представлена желаемая АЧХ НСУ. При нулевом фазовом сдвиге СУ, соответствующая такой АФЧХ, будет являться идеальным фильтром низких частот.

Представим требования к желаемой АФЧХ в виде эталонной модели НСУ $R\{u_r(t), \tilde{y}(t)\}$:

$$\tilde{y}_i(t) = \begin{cases} u_r^i, & \text{если } \omega_i \leq 3,5 \\ 0, & \text{если } \omega_i > 3,5 \end{cases} \quad (3.17)$$

Важной особенностью метода обучения НК с помощью ГА является возможность работы с такими «семантическими» моделями, как (3.17). Вообще, в рамках данного подхода возможно использование любых типов эталонной модели: в виде дифференциальных уравнений, таблич-

ных данных, семантических описаний и прочих. Это в свою очередь позволяет строить широкий спектр НСУ — от системы управления электродвигателем до автоматизированной системы управления технологическим процессом.

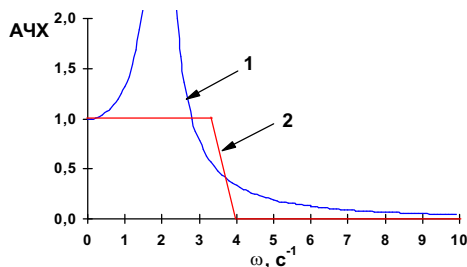


Рис. 3.8

Желаемая амплитудно-частотная характеристика.

Кривая 1 — АЧХ объекта управления;
кривая 2 — желаемая АЧХ нейросетевой системы управления

В соответствии с описанной методикой определим функционал вида (3.4):

$$I(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{\tau} \int_0^{\tau} \left(\frac{y_i(t) - \tilde{y}_i(t)}{|A_i| + \varepsilon} \right)^2 dt, \quad (3.18)$$

Как и в предыдущем подразделе, в функционале (3.18) учитывается непрерывность НСУ, а также нормируется абсолютная величина выходных сигналов. Получаемая квадратичная ошибка между выходами НСУ и эталонной модели усредняется по времени переходного процесса и размеру набора тренировочных сигналов.

Принимая во внимание сделанные в 3.2 выводы, расширим набор тренировочных сигналов (3.12), добавив в него гармонические входные воздействия. Новый набор Σ имеет размер $M=9$ и включает в себя:

$$\begin{aligned}
 u_r^i(t) &= A_i 1(t), \quad A_i = -2 + i, \quad i = \overline{1,3}, \\
 u_r^i(t) &= A_i \sin(\omega_i t), \quad A_i = 1, \quad i = \overline{4,9}, \\
 \omega_4 &= 0,5; \quad \omega_5 = 1; \quad \omega_6 = 2; \quad \omega_7 = 3; \quad \omega_8 = 4; \quad \omega_9 = 5, \\
 t &\in [0, \tau].
 \end{aligned}
 \tag{3.19}$$

Как показывает практика, процесс обучения ПИД-НК при гармонических сигналах задания вначале направляется на уменьшение ошибки по амплитуде выходного сигнала, а затем на уменьшение фазового сдвига. При этом длительность переходных процессов в системе для промежуточных вариантов настройки обычно высока. Чтобы функционал (3.18) позволял в таких случаях точнее оценить качество управления, величину τ необходимо выбрать большей, чем в подразделе 3.2. Поэтому длительность тренировочных входных сигналов примем равной $\tau=15$ с.

На рис. 3.9 приведена схема НСУ с эталонной моделью.

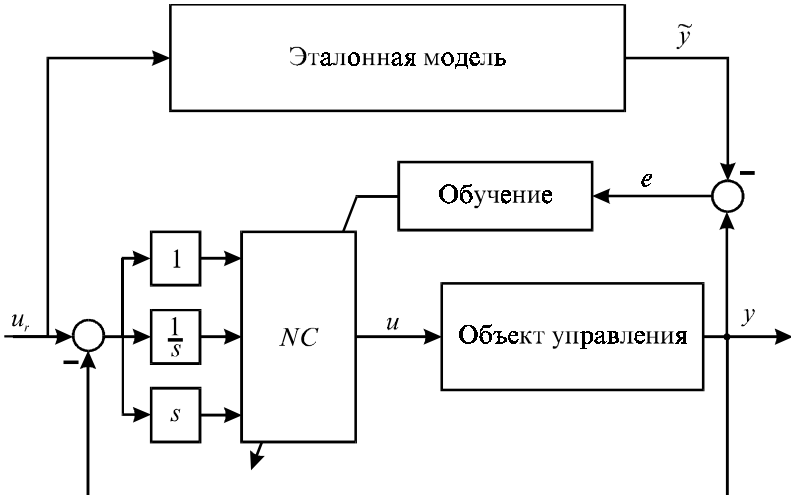


Рис. 3.9

Схема непрерывной нейросетевой системы управления
с эталонной моделью

Учитывая увеличившуюся сложность задачи управления, расширим скрытый слой ИНС и будем использовать в качестве нейросетевой части ПИД-НК МСП структуры 3-10-1. В этом случае вектор проектирования χ имеет вид:

$$\chi = (a_1, w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; a_{10}, w_{0,10}, w_{1,10}, w_{2,10}, w_{3,10}; w_0, w_1, \dots, w_{10}) \quad (3.20)$$

Длина хромосом равна $\lambda_i=10$, $i = \overline{1,61}$.

Параметры полученного в предыдущем примере оптимального решения показали, что область поиска по a_i и w_i можно сузить почти в два раза. Поэтому определим область поиска ГА X следующим образом:

$$a_j \in [0, 5], w_{ij} \in [-1, 1], i = \overline{0,3}, j = \overline{1,10}$$

$$w_i \in [-5, 5], i = \overline{0,10}. \quad (3.21)$$

Увеличение размера набора тренировочных сигналов и их длительности, а также более сложная постановка задачи привели к возрастанию длительности процедуры обучения ИНС. В табл. 3.3, 3.4 приведены значения б1 параметра нейросетевой части обученного ПИД-НК.

Таблица 3.3

Параметры нейросетевой части ПИД-нейроконтроллера.

Скрытый слой

Номер нейрона	a_i	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	3,7732	0,0244	-0,9863	-0,5073	-0,0929
2	4,5601	-0,2630	-1,0000	-0,7674	0,1574
3	1,1926	0,2336	-0,1261	-0,9902	0,5777
4	4,7507	0,7283	-0,9883	0,4995	0,3783
5	3,7977	0,7537	-0,9453	-0,9472	-0,9707
6	2,3705	0,0088	-0,9687	-0,9550	-0,7537
7	0,7087	-0,7048	0,9374	0,9941	0,2649
8	0,5279	-0,7967	-0,9804	-0,9980	-0,5054
9	4,1838	-0,7752	0,9296	0,8749	0,7576
10	1,1730	-0,6911	0,9961	0,9941	0,1261

Параметры нейросетевой части ПИД-нейроконтроллера.

Выходной нейрон

Параметр	Значение	Параметр	Значение	Параметр	Значение
w_0	0,1965	w_4	-0,1887	w_8	-0,6852
w_1	-1,0000	w_5	-0,3021	w_9	0,1124
w_2	-0,9413	w_6	-0,7517	w_{10}	0,4624
w_3	0,4839	w_7	0,7947	—	—

Синтез ПИД-НК занял около 10 часов вычислений на компьютере класса Pentium-100, за это время было промоделировано более 10000 эпох эволюции популяции из 100 особей, и около 100000 раз был вычислен функционал (3.18).

На рис. 3.10 приведены графики переходных процессов в синтезированной НСУ. Как видно статические характеристики НСУ близки к тем, что были получены в предыдущем примере при скачкообразных входных воздействиях. Однако благодаря введению в тренировочный набор гармонических сигналов, значительно улучшились ее частотные характеристики.

Представленные на рис. 3.11 амплитудно- и фазо-частотная характеристики полученной НСУ с ПИД-НК показывают, что поставленная задача в целом достигнута. Фазовый сдвиг на участке $\omega \leq 3,5 \text{ с}^{-1}$ не превышает 5° (см. рис. 3.11,б), коэффициент усиления системы колеблется в пределах $0,9 \div 1,1$ (см. рис. 3.11,а). Обученный ПИД-НК обладает высоким уровнем обобщения новых данных. Благодаря этому его рабочий диапазон входных воздействий $[-2,2]$ шире, чем тренировочный. Как показывают исследования, ширину диапазона ограничивает только максимально достижимое значение выходного сигнала ПИД-НК. Так амплитуда гармонических сигналов задания с частотой, близкой к соб-

ственной частоте объекта управления $\omega \approx 2 \text{ с}^{-1}$, может лежать в пределах $[0,10]$. Это связано с тем, что амплитуда сигнала управления, обеспечивающего такие колебания на выходе системы, в пять раз меньше амплитуды сигнала задания.

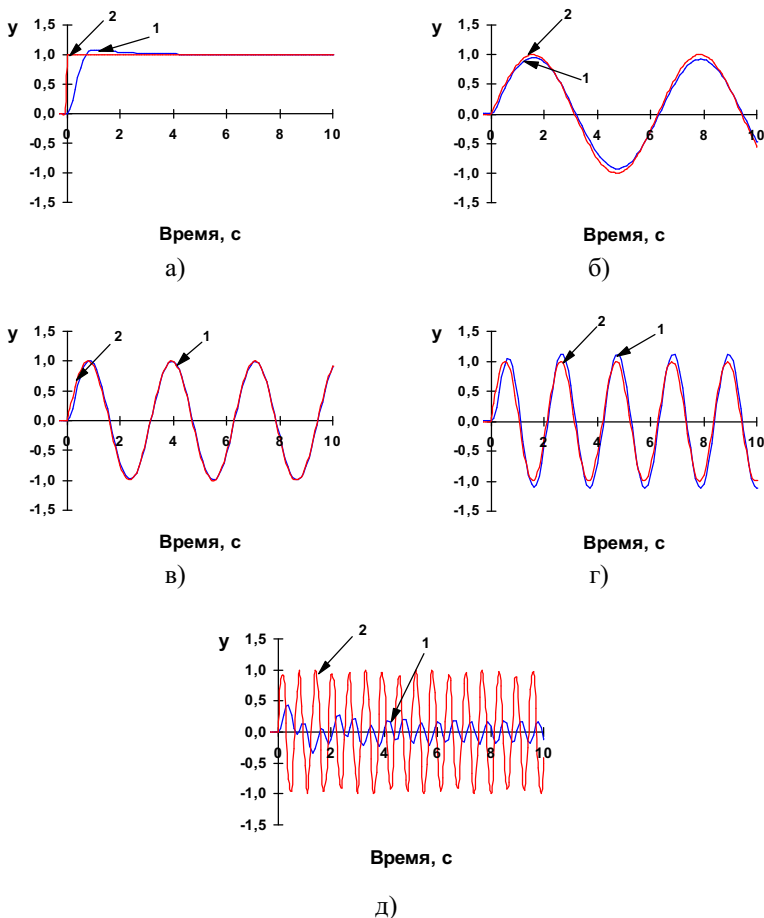


Рис. 3.10

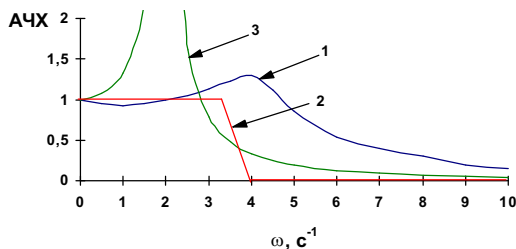
Переходные процессы в НСУ.

кривая 1 — реакция НСУ; кривая 2 — входной сигнал.

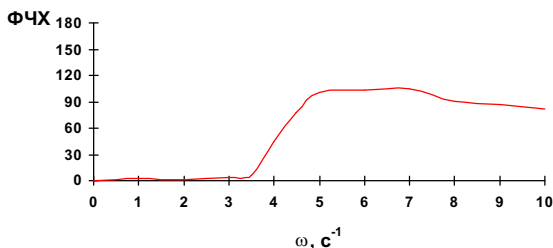
а) — входной сигнал $u_r=1(t)$;

б) — входной сигнал $u_r=\cos(\omega t)$, $\omega=1$; в) — $\omega=2$; г) — $\omega=3$; д) — $\omega=10$

Важно, что ПИД-НК обобщает входные сигналы не только по амплитуде, но и по частоте. Несмотря на дискретный спектр частот тренировочных входных сигналов, использовавшихся при обучении НСУ, ПИД-НК корректно работает и при всех промежуточных частотах. При этом, хотя тренировочные гармонические сигналы имели амплитуду, равную 1, синтезированная НСУ работает с одинаковым качеством для всех амплитуд в рабочем диапазоне.



а)



б)

Рис. 3.11

Амплитудно-частотная и фазово-частотная характеристики НСУ.

а) — амплитудно-частотная характеристика НСУ.

кривая 1 — АФХ синтезированной НСУ; кривая 2 — желаемая АФХ;

кривая 3 — АФХ объекта управления.

б) — фазово-частотная характеристика НСУ

НСУ недостаточно отфильтровывает входные сигналы с частотой $\omega > 3,5 \text{ с}^{-1}$, в этой области ее АФХ отличается от желаемой. Фазово-частотная характеристика объекта при $\omega > 3,5 \text{ с}^{-1}$ также значительно ухудшается. Однако выходной сигнал НСУ во всем диапазоне частот, в отличие от предыдущего примера, не содержит дополнительных гармоник. ПИД-НК не вносит шумов в выход системы, его сигнал управления гладок и непрерывен.

Указанные недостатки объясняются скорее ограничениями самого ПИД-закона управления, чем недотренированностью ИНС. Полученные результаты показывают способность ПИД-НК обобщать в частотной области и реализовывать желаемые АФЧХ. Это делает возможным построение на их базе реальных контроллеров динамических объектов.

Как и в предыдущем случае, линейность тестового объекта никаким образом не учитывалась в процессе синтеза НСУ. Поэтому описанная методика может быть использована и для нелинейных динамических объектов.

3.4. Синтез дискретной нейросетевой системы управления

В предыдущих подразделах были рассмотрены примеры синтеза непрерывных НСУ на базе ПИД-НК. Как было показано, интегрирование ИНС в рамки классических подходов управления расширяет возможности последних. Тем не менее, наибольшие преимущества дает самостоятельное использование ИНС для синтеза дискретных СУ [84, 85].

В рамках таких систем с помощью только ИНС и элементов задержки можно реализовать любой закон управления. Вначале, в зависимости от сложности и порядка динамического объекта выбираются входящие в закон (1.25) глубины задержек по выходу l_1 и входу l_2 объ-

екта. Тем самым определяется структура НСУ. Затем система адаптируется к объекту и внешним воздействиям, формируя в ходе обучения НК необходимый закон преобразования входных сигналов в сигнал управления.

Описанная процедура синтеза дискретной НСУ является универсальной для различных типов объектов управления. Не уменьшая общности, рассмотрим ее подробнее на примере того же тестового динамического объекта (3.10), что использовался в предыдущих подразделах.

Для неизвестного объекта задача определения l_1 и l_2 не имеет однозначного решения. На практике l_1 и l_2 подбираются в ходе нескольких попыток, причем вначале их значения выбираются достаточно большими, чтобы гарантированно обеспечить необходимые динамические свойства НСУ. Однако чрезмерное резервирование приводит к избыточности НК, и требует увеличения размера скрытого слоя МСП. Это негативно сказывается на скорости и качестве обучения НК, поэтому необходимо стремиться к тому, чтобы структура закона управления была минимальной. Тем не менее, если в сигналах обратной связи присутствуют шумы, величина l_1 должна выбираться несколько избыточной, чтобы НСУ могла отфильтровывать помехи.

В данном случае, учитывая порядок объекта управления, примем $l_1=1$ и $l_2=0$. Тогда в общем виде реализуемый НК закон управления описывается следующей зависимостью:

$$u(k) = NC(y(k), y(k-1), u_r(k)). \quad (3.22)$$

На рис. 3.12 представлена схема НСУ на базе соответствующего (3.22) НК. На его входной слой из трех нейронов поступают сигнал внешнего задания $u_r(k)$ и сигнал обратной связи по выходу объекта $y(k)$ на k -том шаге, а также задержанный сигнал обратной связи $y(k-1)$ на предыдущем шаге.

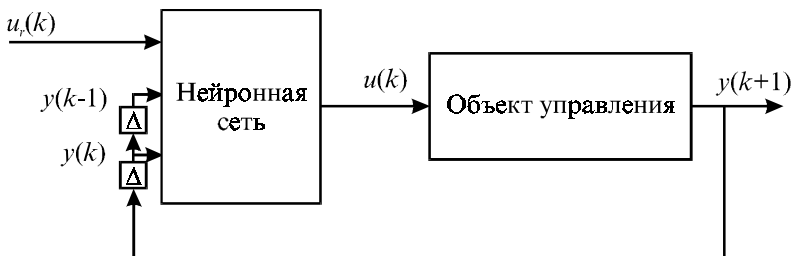


Рис. 3.12

Схема дискретной нейросетевой системы управления

Как и ранее, будем использовать для построения НК трехслойный МСП. Размер скрытого слоя выберем равным 5. Выходной слой НК содержит один линейный нейрон, формирующий сигнал управления на k -том шаге $u(k)$.

Величина шага дискретизации сигнала управления Δ зависит от динамических свойств самого объекта управления и от желаемого рабочего диапазона частот входного воздействия. В данном случае дискретность управления выберем равной:

$$\Delta = 0,1 \text{ с.} \quad (3.23)$$

Сформулируем задачу управления так же, как и в подразделе 3.2. То есть, целью обучения НК является получение линейной следящей СУ. Как и ранее, линейность тестового объекта не влияет на процедуру синтеза, поэтому описываемые подходы являются общими для всех типов динамических систем.

Далее, следуя описанной в 3.1 методике, необходимо сформировать тренировочный набор сигналов Σ . Учитывая, что основная задача примера — показать возможность синтеза дискретной НСУ с НК, выберем для экономии времени набор достаточно компактным: $M=6$:

$$\begin{aligned} u_r^i(k) &= A_i 1(k), \quad A_i = -1,5 + 0,5i, \quad i = \overline{1,5}, \\ u_r^6(k) &= \sin(2k\Delta), \quad t \in [0, \tau]. \end{aligned} \quad (3.24)$$

Здесь $1(k)$ — дискретный аналог единичной функции Хевисайда:

$$1(k) = \begin{cases} 0, & k < 0 \\ 1, & k \geq 0 \end{cases} \quad (3.25)$$

Скачкообразные входные сигналы в тренировочном наборе, как и ранее, необходимы для обеспечения малой статической ошибки НСУ. Введенный дополнительно гармонический сигнал, как показывает опыт, позволяет получать более быстрые переходные процессы. Если такой сигнал не использовать, то синтезируемая НСУ будет иметь колебательную и медленно сходящуюся переходную характеристику. Это — особенность данной схемы НСУ. ПИД-НК, в частности, не требует такой защитной меры.

Частота тренировочного гармонического сигнала выбрана, близкой к собственной частоте объекта управления $\omega \approx 2 \text{ с}^{-1}$. Это позволяет подавить резонансные явления и сделать НСУ близкой к фильтру низких частот.

Длительность тренировочных входных сигналов выберем равной $\tau = 10 \text{ с}$. Качество работы НСУ будем оценивать по функционалу вида (3.5):

$$I(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left(\frac{y_i(k) - u_r^i(k)}{|A_i| + \varepsilon} \right)^4, \quad (3.26)$$

Здесь ошибка между задающим сигналом и выходом объекта управления возводится не в квадрат, как обычно принято, а в 4-ю степень. Дело в том, что использование квадратичных оценок в классических методах проектирования СУ, связано с удобством их аналитического дифференцирования. В то же время, для успешного решения задачи (3.1, 3.2) важно не это, а вид поверхности целевой функции оптимизации и ее способность корректно отображать поставленную задачу управления. Повышая степень ошибки, можно направить поиск прежде

всего против тех решений, которые вызывают сильные расхождения выходного сигнала и сигнала задания. Как показывает опыт, такие решения обладают низкой чувствительностью к изменениям параметров НК, поэтому они тормозят работу ГА, образуя области локальных экстремумов.

Вообще выбор степени, в которую возводится ошибка, зависит от конкретной задачи. Так например ПИД-НК, рассмотренный в предыдущих подразделах, может быть успешно обучен и при нулевой степени, то есть используя в функционалах (3.13, 3.18) модуль ошибки выходного сигнала.

Результаты предыдущих примеров позволили точно оценить область поиска параметров НК, поэтому в данном случае можно отказаться от применения параметра тангенциальной активационной функции a и сократить размерность поискового пространства. Таким образом, параметры НК описываются следующим вектором проектирования χ :

$$\chi = (w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; w_{0,5}, w_{1,5}, w_{2,5}, w_{3,5}; w_0, w_1, w_2, w_3, w_4, w_5) \quad (3.27)$$

Далее определим область поиска X :

$$w_{ij} \in [-1, 1], \quad i = \overline{0,3}, \quad j = \overline{1,5}$$

$$w_i \in [-10, 10], \quad i = \overline{0,5}. \quad (3.28)$$

Длина каждой хромосомы выбирается равной 14-и битам. Это больше, чем в предыдущих примерах, потому что такой тип НСУ требует более точного определения его параметров.

Для оценки приспособленности особи уравнение осциллятора (3.10) преобразуем в форму Коши. Так как данная НСУ является дискретной, а для ее моделирования используется система дифференциальных уравнений, то выход НК рассчитывается только каждые 0,1 с и фиксируется на период дискретизации Δ . Интегрирование системы

осуществляется явным методом Адамса 4-го порядка с постоянным шагом 0,01 с.

Фиксация выхода НК должна проводиться не только при моделировании, но и при реальной работе такой НСУ. Тем не менее, на базе НК могут создаваться и импульсные НСУ, в которых амплитудно модулированный выходной сигнал НК действует на вход объекта управления ограниченное время.

Для завершения синтеза НСУ выберем размер популяции равным 100 и проведем настройку параметров МСП с помощью ГА.

Синтез НСУ занял около 5 часов вычислений на компьютере класса Pentium-100. Оптимальное решение было найдено после 10000 эпох эволюции популяции. В табл. 3.5, 3.6 приведены параметры полученного НК.

Таблица 3.5

Параметры нейроконтроллера. Скрытый слой

Номер нейрона	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	0,1989	0,5797	-0,9609	0,5347
2	-0,2478	-0,2454	0,9433	-0,6090
3	0,1999	-0,7576	0,7498	0,1281
4	0,2038	-0,4604	-0,2493	0,8592
5	-0,0269	-0,0244	-0,5230	0,7810

Таблица 3.6

Параметры нейроконтроллера. Выходной нейрон

Параметр	Значение	Параметр	Значение	Параметр	Значение
w_0	-0,0978	w_2	-3,3920	w_4	5,5034
w_1	1,4370	w_3	-9,8827	w_5	6,2463

На рис. 3.13 представлены переходные процессы в синтезированной НСУ. Переходная характеристика имеет перерегулирование 20 %, статическую ошибку менее 0,005 и полтора колебания переходного процесса. Обученный НК имеет достаточный уровень обобщения как по амплитуде входного сигнала, так и по его частоте. Управление корректно для сигналов задания, лежащих в диапазоне $[-1,2; 1,2]$ и имеющих частоту $\omega < 3 \text{ с}^{-1}$.

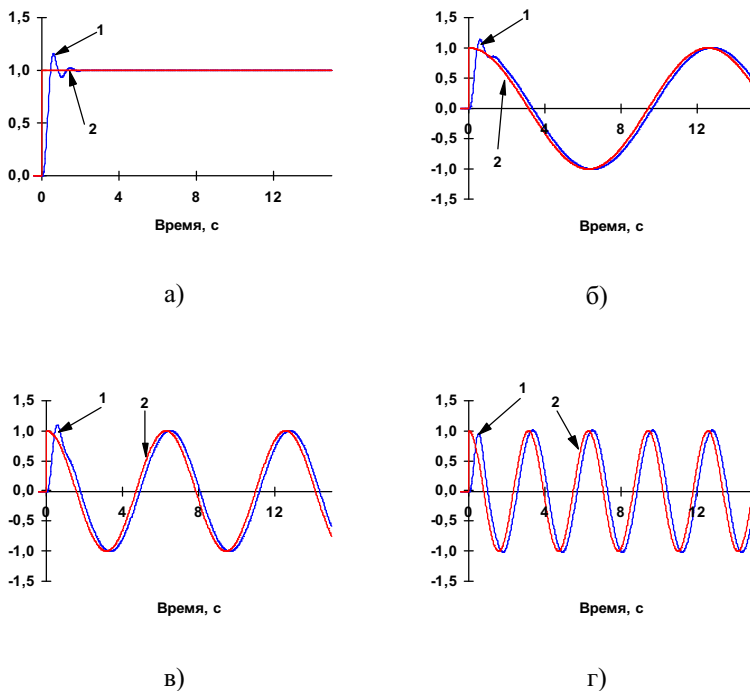


Рис. 3.13

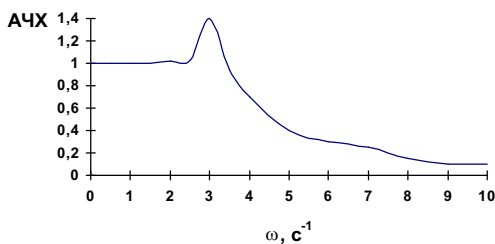
Переходные процессы в НСУ с нейроконтроллером.

Кривая 1 — реакция системы управления;
кривая 2 — входной сигнал.

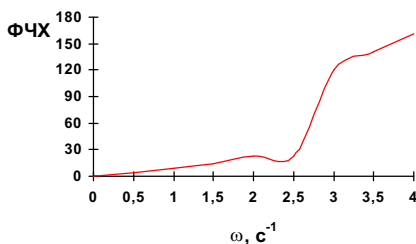
а) — входной сигнал $u_r=1(t)$;

б) — входной сигнал $u_r=\cos(\omega t)$, $\omega=0,5$; в) — $\omega=1$ и г) — $\omega=2$

Синтезированная НСУ имеет переходную характеристику не хуже, чем классические системы с ПИД-НК. Амплитудно-фазовая характеристика НСУ (см. рис. 3.14,а) показывает, что она точно отслеживает гармонические сигналы с частотой $\omega < 2,5 \text{ с}^{-1}$, при этом фазовый сдвиг не превышает 20° (см. рис. 3.14,б). Нейроконтроллеру удалось подавить резонансные явления при собственной частоте объекта. Ограниченное возрастание амплитуды выходного сигнала наблюдается в НСУ уже при более высокой частоте $\omega = 3 \text{ с}^{-1}$.



а)



б)

Рис. 3.14

Амплитудно-частотная и фазо-частотная характеристики системы управления с нейроконтроллером.

- а) — амплитудно-частотная характеристика НСУ;
- б) — фазово-частотная характеристика НСУ

Таким образом, можно заключить, что в процессе обучения ИНС был найден закон управления, обеспечивающий удовлетворительную работу НСУ. Характеристики полученной НСУ в целом не хуже чем у НСУ с ПИД-НК. При этом ее динамические свойства оказались лучше, чем у аналогичной системы с ПИД-НК, также обучавшейся без использования большого количества тренировочных гармонических сигналов.

Приведенный пример доказывает возможность использования предложенной методики для синтеза НК дискретной НСУ. Используя небольшой набор тренировочных сигналов $M < 10$ и МСП с размером скрытого слоя $h \leq 10$, с помощью ГА возможно обучить НК, обеспечивающий НСУ желаемую АЧХ. При этом задача исследователя состоит только в выборе структуры закона управления и подборе тренировочных сигналов.

3.5. Выводы

Полученные в этой главе результаты показывают, что предложенный метод задержанного стимулируемого обучения НК на базе ГА обеспечивает высокий уровень обобщения в ИНС. Введение в вектор проектирования в явном виде параметра, определяющего наклон активационной функции, упрощает процедуру обучения НК.

Скачкообразные тренировочные сигналы гарантируют синтез НСУ с малой статической ошибкой, однако переходная характеристика НСУ, обучаемой только по сигналам вида $u_r = A1(t)$, может быть сильно колебательной. Для того чтобы обеспечить НСУ необходимые динамические характеристики, в тренировочные воздействия необходимо включать гармонические сигналы.

Благодаря универсальным аппроксимационным свойствам ИНС, синтезированные НСУ смогли адаптироваться к объекту управления.

Полученные в ходе обучения конкретные законы управления обеспечивают переходные и частотные характеристики НСУ, не хуже чем у СУ, построенных методами классической теории управления.

НСУ на базе ПИД-НК отличаются высокой скоростью обучения и нулевой статической ошибкой. Эти системы сейчас наиболее выгодны с точки зрения возможности применения в реально действующих технологических комплексах. Реализованные на аналоговых элементах они могут быть быстро и без дополнительных затрат интегрированы, например, в большинство автоматических систем управления, используемых, например, в энергетике.

Дискретные НСУ с НК отличается большей универсальностью. Несмотря на необходимость больших объемов вычислений, такой тип НК позволяет единообразно решать задачи управления разными типами динамических объектов. Используя ГА в качестве метода обучения, можно проводить синтез НСУ только на базе ИНС и элементов задержки.

Тестовый объект, применявшийся в примерах для разработки методики синтеза НСУ, является линейным и хорошо изученным. Это позволило акцентировать внимание на самом процессе синтеза, не усложняя решение. Тем не менее, использовавшиеся подходы являются общими для всех типов объектов управления. Это обосновывается существенной нелинейностью ИНС, и универсальностью ее аппроксимационных свойств.

ГЛАВА 4

ИССЛЕДОВАНИЕ СВОЙСТВ НЕЙРОСЕТЕВЫХ СИСТЕМ УПРАВЛЕНИЯ

Рассмотренные в предыдущей главе примеры синтеза непрерывной и дискретной НСУ доказали эффективность предложенной методики обучения НК с помощью ГА. Однако для того, чтобы признать ее практически применимой, необходимо провести ряд дополнительных исследований.

Благодаря особым свойствам ИНС и глобальности поиска, осуществляемого ГА, разработанная методика является общей для всех типов динамических объектов управления. Тем не менее, остается не выясненным, насколько увеличивается сложность НК для нелинейного объекта. Возможно ли вообще в этом случае задержанное стимулированное обучение НК по тренировочному набору сигналов? Каково необходимое время синтеза, и насколько мощным должен быть компьютер? Ответы на эти вопросы может дать только опыт построения НСУ нелинейным объектом.

В главе 2 было показано, каким образом с помощью ГА на базе ИНС можно построить НЭ динамического объекта. Нейроуправление как парадигма в теории управления подразумевает использование именно нейросетевых моделей объекта управления для синтеза НК. Во всех предыдущих примерах НК обучались только по дифференциальным моделям объекта, поэтому необходимо выяснить, возможно ли применение разработанной методики для синтеза НСУ по НЭ и какими приемами она должна быть дополнена.

Конечной целью при решении задачи управления является создание реальной СУ, способной управлять самим объектом, а не только его моделью. Для того чтобы НСУ могли широко использоваться в про-

мышленности, они должны быть дешевыми, легко настраиваемыми и надежными в эксплуатации. В противном случае можно будет говорить о НСУ только как об уникальных системах, применяемых в единичных случаях. На сегодняшний день немногие применяемые на практике НК реализованы в виде программного обеспечения, и это отталкивает от них внимание разработчиков промышленных СУ. В результате, несмотря на интенсивное развитие теоретических аспектов нейроуправления в последние 20 лет, оно все еще остается на уровне компьютерных экспериментов.

ИНС, как отдельные устройства, обладают рядом свойств, делающих их привлекательными для аппаратной реализации на базе моделей отдельных нейронов. Поэтому одна из важных задач исследования НСУ — выяснить, обладают ли они такими же свойствами, и возможна ли аппаратная реализация НСУ.

Положительные ответы на поставленные вопросы позволяют надеяться, что уже в ближайшие 5—10 лет в мире появятся промышленные образцы НСУ. Также учитывая современную тенденцию к интеллектуализации СУ, можно ожидать, что дальнейшее развитие теории автоматического управления будет связано с созданием на базе симбиоза ИНС и ГА систем искусственного интеллекта, решающих задачи управления сложными технологическими комплексами.

4.1. Синтез нейросетевой системы управления нелинейным объектом

4.1.1. Нейросетевая система управления на базе нейроконтроллера.

Для того чтобы оценить, насколько возрастает сложность задачи синтеза НСУ в случае нелинейного объекта, сделаем один из параметров модели (3.10) зависимым от ее выхода [86]. Учитывая, что использованный тестовый объект — колебательная система, сделаем нелинейным ее коэффициент жесткости ζ , приблизив тем самым пример к реальному случаю.

Новый тестовый нелинейный динамический объект управления описывается следующей системой:

$$\begin{aligned} T^2 x'' + 2T\zeta(x)x' + x &= ku; \\ y &= x, \end{aligned} \quad (4.1)$$

где $T = 0,5$; $k = 1$;

$$\zeta(x) = (|x| - 0,5)^2. \quad (4.2)$$

На рис. 4.1 представлены переходные процессы в объекте для разных амплитуд скачкообразного сигнала задания. Как видно, из-за нелинейности объекта его динамические характеристики достаточно сильно меняются в зависимости от величины выходного сигнала y .

Используя тот же закон управления (3.22), что и в подразделе 3.4, синтезируем линейную следящую НСУ нелинейным объектом (4.1). Таким образом, $l_1=1$, $l_2=0$, и

$$u(k) = NC(y(k), y(k-1), u_r(k)) \quad (4.3)$$

Схема синтезируемой НСУ представлена на рис. 3.12.

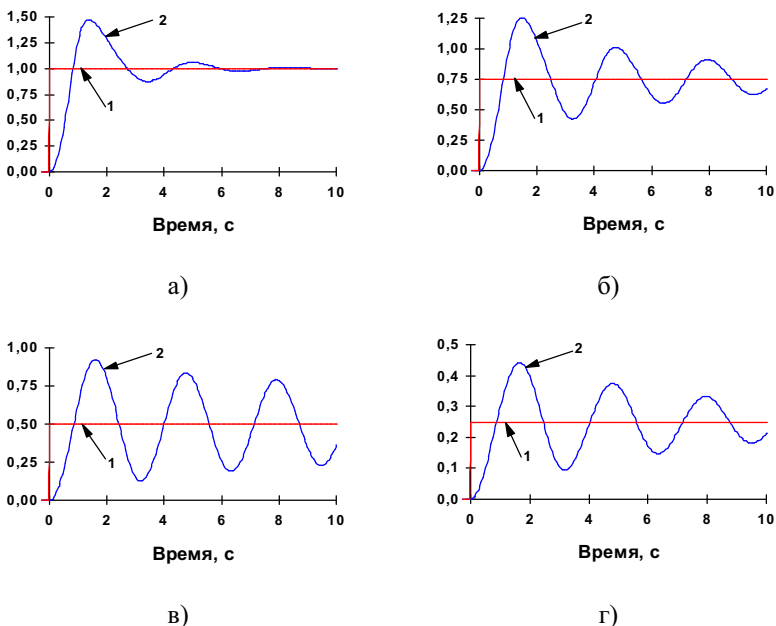


Рис. 4.1

Переходные процессы в нелинейном объекте управления.

Кривая 1 — сигнал задания; кривая 2 — реакция НСУ.

Входной сигнал $u=A1(t)$.

а) — $A=1$; б) — $A=0,75$; в) — $A=0,5$; г) — $A=0,25$

Важно, что рассматриваемая НСУ линейна только в смысле характеристик вход-выход, но по своей сути является нелинейной, так как состоит из существенно нелинейных элементов.

Для построения НК выберем МСП структуры 3-10-1 с тангенциальной активационной функцией нейронов скрытого слоя. В предыдущем примере НК содержал лишь 5 скрытых нейронов, однако в данном случае объект более сложный и, соответственно, должен усложниться закон управления (4.3).

Аппроксимирующие свойства ИНС непосредственно связаны с размером ее скрытого слоя, поэтому для расширения возможностей НК

необходимо увеличивать количество его скрытых нейронов. Однако, как и для РБФ-сетей, простое наращивание структуры сети при сохранении такого же количества тренировочных сигналов может привести к ее чрезмерной избыточности. При этом неизбежно пострадает способность ИНС обобщать новые данные. Для рассматриваемой задачи 10 скрытых нейронов НК являются достаточными. Число это было определено в ходе серии экспериментов при последовательном увеличении размера скрытого слоя, начиная с 5. Заранее точно определить его на сегодняшний день невозможно, реально лишь оценить необходимую структуру сети, исходя из предшествующего опыта [87, 88].

На входной слой НК поступают сигнал внешнего задания $u_r(k)$, сигнал обратной связи по выходу объекта $y(k)$ на k -том шаге, а также задержанный сигнал обратной связи $y(k-1)$ на предыдущем шаге. Выходной нейрон, формирует сигнал управления на k -том шаге $u(k)$. Шаг дискретизации НСУ, как и в подразделе 3.4, положим равным $\Delta = 0,1$ с.

Для обучения НК будем использовать скачкообразные тренировочные сигналы. Так как свойства нелинейного объекта управления меняются с величиной выходного сигнала, необходимо выбрать более мелкий шаг изменения амплитуды тренировочных сигналов, чем для линейного объекта. В противном случае сеть не сможет построить требуемую аппроксимационную поверхность, а значит, уровень обобщения в ней будет недостаточным для корректной работы НСУ.

Таким образом, тренировочный набор сигналов Σ имеет размер $M=9$ и включает в себя следующие сигналы:

$$u_r^i(k) = A_i 1(k), \quad A_i = -1,25 + 0,25i, \quad i = \overline{1,9}, \quad t \in [0, \tau]. \quad (4.4)$$

Скачкообразные тренировочные сигналы гарантируют синтез НСУ с малой статической ошибкой, однако переходная характеристика НСУ, обучаемой только по сигналам вида $u_r=A1(t)$ может быть сильно колебательной. Для того чтобы обеспечить НСУ необходимые динамические

характеристики, в тренировочный набор можно добавить гармонический сигнал, как это было сделано в предыдущем примере, или модифицировать функционал качества, согласно которому ведется обучение НК.

Здесь выгоднее применить второй прием, чтобы не увеличивать размер тренировочного набора сигналов, а значит, и время синтеза НСУ. Включим в функционал качества кроме интегральной ошибки управления по выходному сигналу, также интегральную ошибку по его производной. Соответствующий функционал качества имеет вид:

$$I(y) = I_1(y) + \beta I_2(y). \quad (4.5)$$

Где

$$I_1(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left(k \frac{y_i(k) - u_r^i(k)}{|A_i| + \varepsilon} \right)^2, \quad (4.6)$$

$$I_2(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left(k \frac{y_i(k) - y_i(k-1) - u_r^i(k) + u_r^i(k-1)}{(|A_i| + \varepsilon)\Delta} \right)^2. \quad (4.7)$$

Величина веса β выбирается такой, чтобы уравнивать вклады в результирующий функционал I ошибок управления по выходу I_1 и его производной I_2 :

$$\beta = 0,05. \quad (4.8)$$

Длительность тренировочных входных сигналов примем равной $\tau = 10$ с.

Кроме того, что (4.5) является сверткой, от (3.26) его отличает введение в (4.6) и (4.7) в качестве множителя времени, представленного номером шага работы НСУ k .

Как было указано ранее, вид поверхности целевой функции оказывает решающее влияние на результаты поиска. Несмотря на то, что функционалы вида (3.26) и (4.5) имеют одинаковое оптимальное значение $I=0$, соответствующее идеальной следящей СУ, получаемые с их

помощью решения обладают разными свойствами. Различие состоит не столько в виде переходных процессов, сколько в приемлемости полученной НСУ. Опыт показывает, что при использовании функционала (3.26) процесс обучения НК идет длительное время. Поверхность целевой функции в этом случае покрыта множеством локальных минимумов, вследствие чего достижение области глобального оптимума возможно лишь с помощью популяции размером более чем 150 особей. Однако и после стягивания популяции вблизи оптимума, получаемое решение не стабилизируется. То есть даже незначительное улучшение функционала качества может быть связано не с уточнением глобально решения, а с переходом поисковой точки в соседний локальный минимум. При этом существенно меняется вид переходных процессов в НСУ: либо низкая колебательность при большой статической ошибке, либо малая ошибка при высокой колебательности. В результате, даже после 50000 эпох эволюции характеристики НСУ остаются неудовлетворительными. Функционал вида (4.5) позволяет избежать такой ситуации и быстро получить настройку НК, обеспечивающую высокое качество управления.

Будем использовать при обучении НК в явном виде параметр тангенциальной активационной функции a . Тогда вектор проектирования НК χ имеет вид:

$$\chi = (a_1, w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; a_{10}, w_{0,10}, w_{1,10}, w_{2,10}, w_{3,10}; w_0, w_1, \dots, w_{10}). \quad (4.9)$$

Область поиска оптимальных параметров НК X выберем следующей:

$$a_j \in [0, 3], \quad w_{ij} \in [-1, 1], \quad i = \overline{0, 3}, \quad j = \overline{1, 10}; \quad w_i \in [-10, 10], \quad i = \overline{0, 10}. \quad (4.10)$$

При этом длина каждой хромосомы полагается равной 14-и битам.

Для моделирования работы НСУ дифференциальное уравнение объекта управления (4.1) преобразуется в форму Коши. Интегрирование

осуществляется явным методом Адамса 4-го порядка с постоянным шагом 0,01 с.

В табл. 4.1, 4.2 приведены параметры НК, полученного после 10000 эпох эволюции популяции из 100 особей. Время его обучения составило около 7-и часов вычислений на компьютере класса Pentium-100.

Таблица 4.1

Параметры нейроконтроллера. Скрытый слой

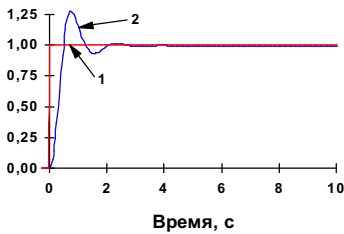
Номер нейрона	a	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	2,4340	0,9394	0,8768	-0,0147	0,9453
2	0,9501	0,3920	0,1241	-0,7517	-0,0010
3	0,8915	0,4995	-0,3783	0,7498	-0,5015
4	1,7067	-0,0518	0,2023	-0,7595	0,8123
5	0,4252	0,5777	0,2981	-0,6266	0,6891
6	2,2639	0,0753	0,6520	-0,8807	0,5484
7	0,1935	0,0616	0,8768	-0,3763	-0,1261
8	1,0587	-0,9238	-0,2493	-1,0000	-1,0000
9	0,2229	-0,7615	0,5621	-0,5073	0,5152
10	0,5513	-0,4545	-0,9062	-0,6246	0,4858

Таблица 4.2

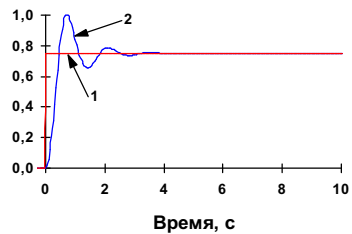
Параметры нейроконтроллера. Выходной нейрон

Параметр	Значение	Параметр	Значение	Параметр	Значение
w_0	-0,0694	w_4	1,5934	w_8	0,7625
w_1	0,5816	w_5	2,4682	w_9	4,3939
w_2	1,5054	w_6	3,7048	w_{10}	4,1349
w_3	0,3959	w_7	3,8661	—	—

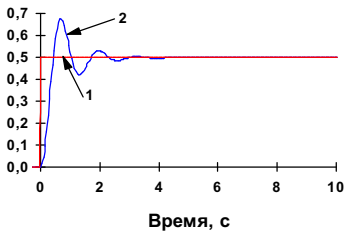
На рис. 4.2 представлены переходные процессы в синтезированной с помощью ГА НСУ.



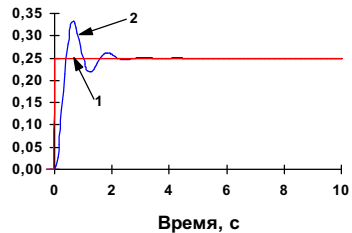
а)



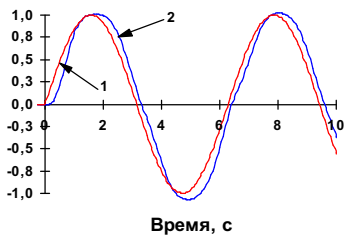
б)



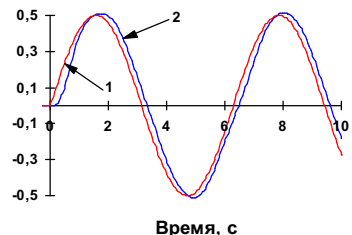
в)



г)



д)



е)

Рис. 4.2

Переходные процессы в синтезированной НСУ.

Кривая 1 — сигнал задания; кривая 2 — реакция НСУ.

Входной сигнал $u_r = A1(t)$. а) — $A=1$; б) — $A=0,75$;

в) — $A=0,5$; г) — $A=0,25$.

Входной сигнал $u_r = A\sin(t)$. д) — $A=1$; е) — $A=0,5$

Несмотря на нелинейность объекта управления в рамках того же закона, что использовался для линейного тестового объекта, был получен НК, обеспечивающий практически одинаковое качество управления

во всем рабочем диапазоне входных сигналов $u_r \in [-1,1]$. Переходные процессы в НСУ при скачкообразных входных сигналах различной амплитуды имеют перерегулирование не более 30 %, статическую ошибку менее 0,005 при полутора— двух колебаниях (см. рис. 4.2,а,б,в,г). Обученный НК обладает достаточным уровнем обобщения не только по амплитуде входного сигнала, но и по частоте (см. рис. 4.2,д,е). Его управление корректно для сигналов задания имеющих частоту $\omega < 3 \text{ с}^{-1}$.

На основе полученного НК можно продемонстрировать еще одно свойство НСУ. Предположим, что в канале обратной связи по выходу объекта управления присутствуют шумы, обусловленные его собственными свойствами или внешней помехой. На рис. 4.3 представлены переходные процессы в синтезированной НСУ при случайной помехе с равномерным законом распределения плотности вероятности и амплитудой, составляющей 10 % от амплитуды сигнала задания. Как показывают графики, НК существенно сглаживает эти шумы, а отклонение выхода НСУ от сигнала задания не превышает 15 %.

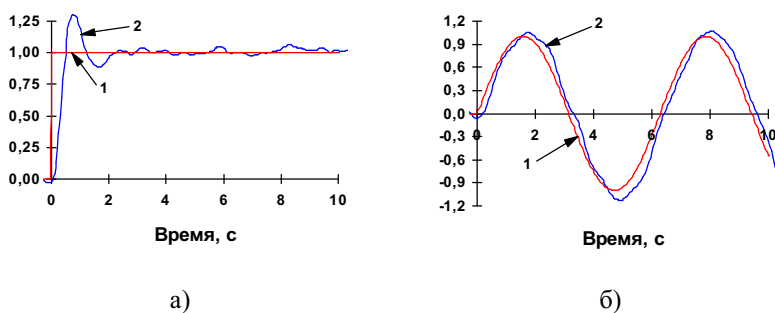


Рис. 4.3

Переходные процессы в НСУ при наличии шума
в сигнале обратной связи.

Кривая 1 — сигнал задания; кривая 2 — реакция НСУ.

Входной сигнал а) — $u_r=1(t)$; б) — $u_r=\sin(t)$

4.1.2. Синтез ПИД-нейроконтроллера для управления температурой пара на выходе котла ТГМП-204.

СУ на базе П-, ПИ- и ПИД-контроллеров с постоянными значениями уставок остаются на сегодняшний день самым распространенным типом промышленных систем автоматического управления, в частности, в энергетике. Это определяется как удовлетворительным качеством работы таких СУ, так и сложностью синтеза более сложных систем на базе контроллеров высокого порядка.

Одним из важнейших элементов комплексной системы автоматического управления энергоблоком тепловой станции, является система регулирования температуры перегретого пара на выходе парового котла (см. рис. 4.4).

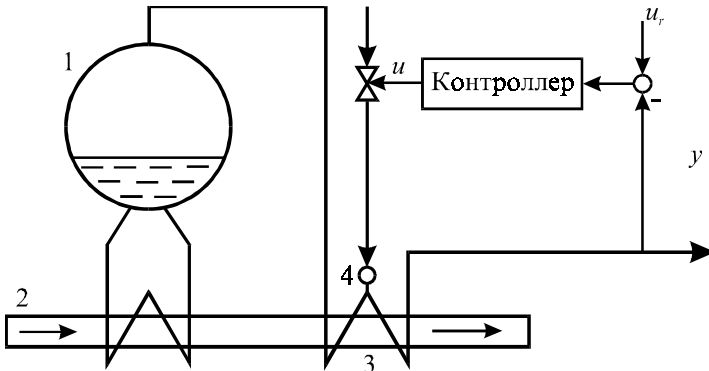


Рис. 4.4

Упрощенная схема системы регулирования температуры перегретого пара на выходе парового котла.

- 1 — барабан; 2 — газоход; 3 — пароперегреватель;
- 4 — впрыскивающий пароохладитель

Котел представляет собой нелинейный динамический объект управления, та как его параметры существенно зависят от нагрузки энергоблока. Так для парового котла ТГМП-204 энергоблока мощно-

стью 800 МВт в ходе работы нагрузка может реально изменяться в пределах от 400 до 800 МВт. Задача СУ — обеспечивать устойчивую работу котла во всем этом диапазоне.

В [89] приведена передаточная функция котла ТГМП-204 по возмущению впрыском, синтезированная по экспериментальным динамическим характеристикам:

$$W(s) = \frac{a_0 e^{-\tau s}}{1 + b_1 s + b_2 s^2}. \quad (4.11)$$

В табл. 4.3 представлены параметры модели (4.11), а на рис.4.5 — ее переходные функции для режимов нагрузки 400, 600 и 800 МВт.

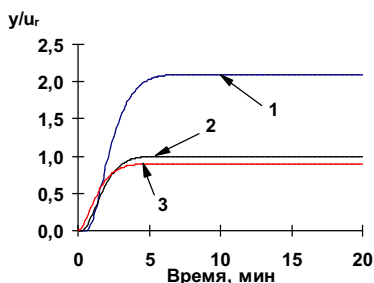


Рис. 4.5

Переходные характеристики котла ТГМП-204 при различных нагрузках энергоблока.

1 — 400 МВт; 2 — 600 МВт; 3 — 800 МВт

Таблица 4.3

Параметры передаточной функции котла ТГМП-204 по каналу регулирования температуры перегретого пара при различных режимах нагрузки

Режим	Мощность, МВт	τ	a_0	b_1	b_2
1	400	0,605	2,091	1,815	1,027
2	600	0,264	0,996	1,543	0,784
3	800	0,000	0,896	1,415	0,613

Используя данную модель, применим предложенную в предыдущей главе методику для построения на базе ПИД-НК непрерывной НСУ температурой перегретого пара на выходе котла ТГМП-204.

Сформируем следующий тренировочный набор сигналов Σ размером $M=3$:

$$u_r^i(t) = A_i 1(t), \quad t \in [0, \tau],$$

$$A_i = -2 + i, \quad i = \overline{1,3}. \quad (4.12)$$

Для оценки качества управления, реализуемого ПИД-НК, будем использовать сумму функционалов вида (3.13), рассчитанных для каждого из трех режимов энергоблока:

$$I(y) = \sum_{k=1}^3 \left\{ \frac{1}{M} \sum_{i=1}^M \frac{1}{\tau} \int_0^{\tau} \left(\frac{y_i(t) - u_r^i(t)}{|A_i| + \varepsilon} \right)^2 dt \right\}_k, \quad (4.13)$$

где индекс k означает, что блок работает в k -том режиме нагрузки.

Длительность тренировочных входных сигналов примем равной $\tau=20$ с.

В качестве нейросетевой части ПИД-НК выберем МСП структуры 3-10-1. Соответствующий вектор проектирования ПИД-НК χ имеет вид:

$$\chi = (a_1, w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; a_{10}, w_{0,10}, w_{1,10}, w_{2,10}, w_{3,10}; w_0, w_1, \dots, w_{10}). \quad (4.14)$$

Длину каждой хромосомы положим равной $\lambda_i=10$, $i = \overline{1,61}$ и будем проводить поиск оптимальных параметров ПИД-НК в области X :

$$a_j \in [0, 0,5], \quad w_{ij} \in [-1, 1], \quad i = \overline{0,3}, \quad j = \overline{1,10}$$

$$w_i \in [-2, 2], \quad i = \overline{0,10}. \quad (4.15)$$

Для моделирования работы НСУ по передаточной функции объекта управления (4.11) строится система дифференциальных уравнений

в форме Коши. Интегрирование осуществляется явным методом Адамса 4-го порядка с постоянным шагом 0,01 с.

В табл. 4.4, 4.5 приведены значения параметров нейросетевой части обученного ПИД-НК. Его синтез занял около 15 часов вычислений на компьютере класса Pentium-100, за это время было промоделировано более 10000 эпох эволюции популяции из 100 особей, и около 100000 раз был вычислен функционал (4.13).

Таблица 4.4

Параметры нейросетевой части ПИД-нейроконтроллера.

Скрытый слой

Номер нейрона	a_i	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	0,0802	0,9648	-0,0342	0,9980	1,0000
2	0,0850	-0,6774	-0,0010	0,9433	-0,9746
3	0,1569	0,6520	0,9980	-0,0029	0,9883
4	0,4184	0,6598	0,1241	0,1730	0,9707
5	0,0303	0,1417	1,0000	-0,9687	-0,9394
6	0,1916	-0,5249	1,0000	0,8065	0,9844
7	0,4702	-0,4154	0,9941	0,5308	-0,9726
8	0,1872	0,4018	1,0000	0,7576	1,0000
9	0,0958	0,4585	1,0000	0,9707	-0,0010
10	0,1271	-0,4976	0,9980	-0,0029	0,8690

Таблица 4.5

Параметры нейросетевой части ПИД-нейроконтроллера.

Выходной нейрон

Параметр	Значение	Параметр	Значение	Параметр	Значение
w_0	-0,0674	w_4	0,8622	w_8	0,4927
w_1	0,3734	w_5	1,0381	w_9	1,3196
w_2	0,3715	w_6	1,4545	w_{10}	0,8954
w_3	0,5904	w_7	0,6706	—	—

Помимо модели котла в [89] также предложены параметры робастного ПИД-контроллера, нечувствительного к изменению режима мощности энергоблока:

$$W_{\text{пид}}(s) = 2,221 \frac{1 + 0,253s}{1 + 0,002s} \frac{1 + 1,4s + 0,501s^2}{(1 + 0,001s)2,199s}. \quad (4.16)$$

На рис. 4.6,а приведены переходные процессы в синтезированной НСУ на базе ПИД-НК для каждого из трех режимов работы энергоблока: 400, 600 и 800 МВт. На рис. 4.6,б представлены переходные процессы в тех же режимах нагрузки, но для СУ с ПИД-контроллером (4.16).

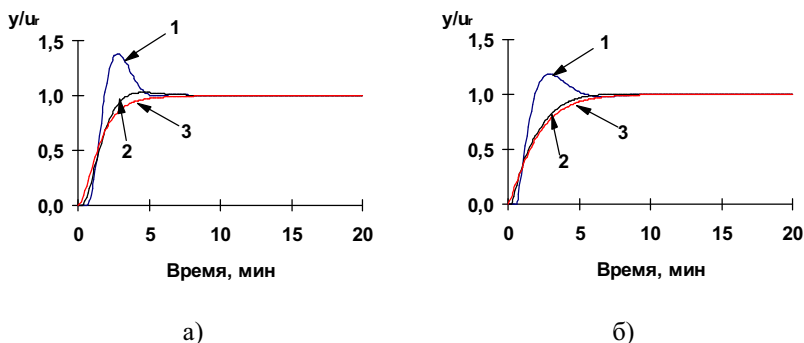


Рис. 4.6

Переходные процессы на выходе котла ТГМП-204.

а) — для НСУ на базе ПИД-нейроконтроллера;

б) — для СУ на базе ПИД-контроллера.

Нагрузка блока: кривая 1 — 400 МВт; кривая 2 — 600 МВт;
кривая 3 — 800 МВт

Как видно, НСУ справилась с поставленной задачей не хуже, чем СУ с ПИД-контроллером. В табл. 4.6 представлены значения интегральной оценки качества переходных характеристик в обеих системах, для всех режимов нагрузки.

Сравнительные характеристики систем управления котла ТГМП-204
на базе ПИД-контроллера и ПИД-нейроконтроллера

Режим	Квадратичная интегральная оценка качества	
	СУ с ПИД-контроллером	НСУ с ПИД-нейроконтроллером
1	3,57	3,01
2	2,35	1,91
3	2,59	2,35

Закон управления, реализуемый ПИД-НК, обеспечивает системе во всех режимах нагрузки быстродействие на 10 % большее, чем при использовании традиционного ПИД-контроллера, при этом фазовый сдвиг в НСУ для всех режимов меньше. Перерегулирование в НСУ при нагрузке блока 400 МВт больше чем у СУ с ПИД-контроллером (4.16), однако не превышает 40 %. При работе блока с нагрузкой 600 МВт полученный в [89] ПИД-контроллер почти вдвое снижает быстродействие системы по сравнению с характеристиками самого объекта. Синтезированный ПИД-НК, благодаря нелинейности его нейросетевой части, не страдает таким недостатком.

Как положительную черту НСУ можно отметить, что генерируемый ПИД-НК сигнал управления более гладкий, чем у ПИД-контроллера, и ограничен по амплитуде на уровне 2,5. В то время как сигнал управления ПИД-контроллера резко меняется и содержит неограниченные всплески.

В целом, приведенные в этом подразделе примеры доказывают правомочность сделанного в предыдущей главе утверждения. Разработанная методика синтеза НСУ с помощью ГА является универсальной для всех типов объектов управления. Нелинейность ИНС и их уникальные аппроксимационные свойства теоретически позволяют решать в

рамках нейроуправления задачи любого уровня сложности. При этом структура НК и время его обучения не увеличиваются катастрофически с возрастанием сложности объекта управления. Для большинства практических задач управления верхнюю границу сложности НК можно оценить как 20—25 нейронов входного слоя и 50—60 нейронов скрытого слоя. А это значит, что для синтеза НСУ вполне достаточно персонального компьютера и 50—100 часов машинного времени.

В заключение можно провести сравнение результатов, полученных в подразделах 3.2—4.1. В табл. 4.7 приведена обобщенная сравнительная характеристика синтезированных ПИД-НК и НК.

Результаты показывают, что главным преимуществом НСУ на базе ПИД-НК является легкая интегрируемость в действующие непрерывные СУ, в то время как дискретные НСУ отличаются большей универсальностью и широтой круга решаемых задач.

Таблица 4.7

Сравнительная характеристика ПИД-нейроконтроллера
и нейроконтроллера

Характеристика	ПИД-нейроконтроллер	Нейроконтроллер
Закон управления	непрерывный	дискретный
Размер тренировочного набора сигналов	небольшой	большой
Целевая функция задачи обучения	стандартная оценка качества управления	модифицированные критерии качества
Длительность процедуры обучения	низкая	высокая
Требуемая точность модели объекта управления	высокая	средняя
Точность решения задачи управления	высокая	достаточная
Сложность реализуемых законов управления	ограниченная	неограниченная
Способ внедрения	модификация действующих СУ	построение новых НСУ

4.2. Синтез нейросетевой системы управления по нейроэмулятору объекта управления

Во всех предыдущих примерах НК обучались при помощи дифференциальных моделей динамических объектов управления. В тоже время в рамках нейроуправления возможно построение точных нейросетевых моделей объектов. Естественно возникает вопрос, возможен ли синтез НСУ только по НЭ, без привлечения другого вычислительного аппарата, кроме ИНС?

Чтобы ответить на этот вопрос, рассмотрим пример построения НСУ осциллятором, использовавшимся в качестве тестового объекта в подразделе 3.4. Однако на этот раз для моделирования переходных процессов в объекте будем использовать не (3.10), а его НЭ (2.31), синтезированный в подразделе 2.4.

Вначале проведем обратный эксперимент, то есть используем полученный подразделе 3.4 НК для управления объектом, представленным в виде НЭ [52]. Как оказывается, получаемые таким образом переходные процессы в НСУ существенно отличаются от тех, что приведены на рис. 3.13. Причина этого состоит в особенностях работы НЭ и НК.

На рис. 4.7 изображены графики сигнала управления u , генерируемого НК, и выход объекта управления, описываемого дифференциальной моделью, при единичном скачке $u_r=1(t)$ на входе НСУ. Как видно, сигнал управления изменяется в пределах $u \in (-3,5)$, в то же время, синтезированный НК корректно работает лишь в диапазоне $u \in [-1,1]$. На рис. 4.8 приведены переходные процессы в объекте, моделируемом с помощью дифференциальных уравнений и НЭ, при входном сигнале $u=4 \cdot 1(t)$. Эти графики сильно различаются.

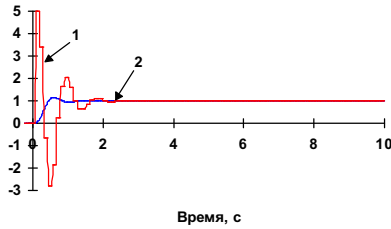


Рис. 4.7

Переходные процессы в НСУ, синтезированной по дифференциальной модели объекта управления.

Кривая 1 — сигнал управления $u_r=1(t)$;
кривая 2 — выход НСУ

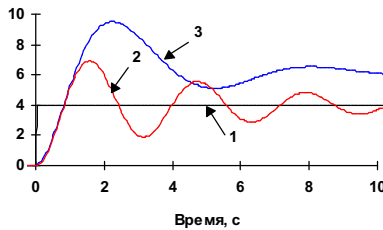


Рис. 4.8

Работа нейроэмулятора при больших значениях амплитуды входного воздействия.

Кривая 1 — входной сигнал;
кривая 2 — реакция объекта управления;
3 — реакция нейроэмулятора

Так как во время моделирования работы НСУ с помощью НЭ объекта управления границы достоверности модели неоднократно нарушались, то в результате были получены переходные процессы, не соответствующие действительности. При синтезе НК по НЭ необходимо учитывать возможность такой ситуации. Если не принять специаль-

ные меры, направленные на соблюдение границ достоверности модели, то обученный НК не сможет управлять реальным объектом, хотя для модели объекта его характеристики будут идеальными.

Как и в примере, приведенном в подразделе 3.4, положим глубины задержек по выходу и входу объекта управления равными:

$$l_1=1, l_2=0. \quad (4.17)$$

Тогда закон управления, реализуемый НК, имеет вид:

$$u(k) = NC(\hat{y}(k), \hat{y}(k-1), u_r(k)), \quad (4.18)$$

где \hat{y} — выходной сигнал НЭ.

На рис. 4.9 представлена схема обучения НК НСУ, с помощью НЭ динамического объекта управления [90, 91].

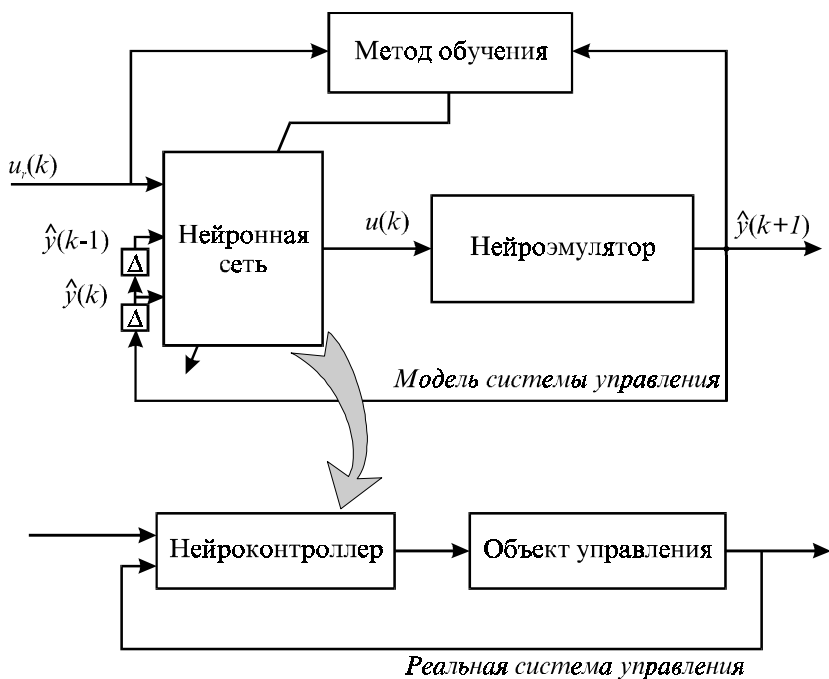


Рис. 4.9

Схема обучения нейроконтроллера по нейроэмулятору

Как и ранее, будем строить НК на базе трехслойного МСП. Однако, в этот раз из-за усложнения задачи размер скрытого слоя положим равный 10.

В данной схеме обучения НСУ имеет два уровня дискретизации. Шаг предсказания НЭ равен $\Delta_e=0,05$, а дискретность сигнала управления НК такая же, как и в подразделе 3.4 — $\Delta_c=0,1$. Таким образом, моделирование переходных процессов в НСУ проходит с шагом дискретизации Δ_e , а сигнал управления вычисляется раз за два шага. В то же время, при использовании НК для управления реальным объектом, дискретность НСУ равна Δ_c .

Для оценки качества работы НСУ и учета границ достоверности НЭ введем функционал в виде свертки:

$$I = I_y + \frac{I_u}{\beta I_y + \varepsilon_y}, \quad (4.19)$$

где

$$I_y = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left(\frac{y_i(k) - u_r^i(k)}{|A_i| + \varepsilon} \right)^4, \quad (4.20)$$

$$I_u = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} Q(u(k))^2, \quad (4.21)$$

$$Q(u(k)) = \begin{cases} |u(k)| - 1, & \text{если } |u(k)| > 1 \\ 0, & \text{если } |u(k)| \leq 1 \end{cases}. \quad (4.22)$$

Здесь $Q(u(k))$ — штрафная функция, увеличивающая функционал I в том случае, когда сигнал управления выходит за область достоверности НЭ. A_i — как и раньше, амплитуда тренировочного входного сигнала u_r^i .

При сворачивании функционалов I_y и I_u в результирующий функционал качества I , величина штрафа меняется в зависимости от значе-

ния I_y . Таким образом, по мере улучшения качества переходных процессов в НСУ, все более усиливаются требования к ограничениям на управление. Малая добавка $\epsilon_y \approx 0,01$ обеспечивает защиту функционала I от гипотетической ситуации $I_y=0$, а весовой коэффициент β служит для нормирования I_y . В данной задаче $\beta=5$.

Несмотря на введенный штраф, ГА может генерировать в ходе обучения варианты НК, сигналы управления которых нарушают границы достоверности НЭ. Опасность состоит в том, что вычисляемая по некорректно работающему НЭ величина I_y , может оказаться намного меньше истинной. Таким образом, процесс поиска может быть направлен по ложному пути и полученное решение окажется недействительным. Чтобы избежать такой ситуации, необходимо ограничивать сигнал управления, поступающий на вход НЭ.

Для этого введем промежуточный сигнал \bar{u} равный:

$$\bar{u} = \begin{cases} 1, & \text{если } 1 < u \\ u, & \text{если } -1 \leq u \leq 1 \\ -1, & \text{если } -1 > u \end{cases} \quad (4.23)$$

и будем проводить моделирование переходных процессов в объекте с помощью его НЭ, но уже по закону:

$$\hat{y}(k+1) = NE(\hat{y}(k), \hat{y}(k-1), \hat{y}(k-2), \hat{y}(k-3), \bar{u}(k)). \quad (4.24)$$

Несмотря на то, что такой прием нарушает точность оценивания качества управления, с помощью функционала I_y , он позволяет избежать полного искажения этой оценки. В то же время, минимизируя штраф I_u , ГА ищет варианты настройки НЭ удовлетворяющие условию:

$$|u(k)| \leq 1, \quad k \in [0, \infty). \quad (4.25)$$

Для них функционал I_y рассчитывается уже корректно. Полученный оптимальный закон управления не должен нарушать условие (4.25), поэтому в реальной НСУ ограничение сигнала управления u не требуется.

Необходимо заметить, что в (4.18) и (4.24) номер шага k имеет разное значение. Для закона управления (4.18) $t_{k+1}=t_k+\Delta_c$, в то же время для НЭ (4.24) $t_{k+1}=t_k+\Delta_e$. Здесь не была введена двойная индексация шагов, чтобы не усложнять изложение, но необходимо помнить, что сигнал управления вычисляется раз в два шага эмулирования, и его дискретность равна Δ_c .

Как и в подразделе 3.4, целью обучения НК является получение линейной следящей СУ. Однако в данном случае, в связи с ограничениями на максимальную амплитуду сигнала управления, рабочий диапазон сигнала задания НСУ u_r необходимо выбрать более узким:

$$u_r \in [-0,5; 0,5]. \quad (4.26)$$

Учитывая (4.26) сформируем тренировочный набор сигналов Σ размером $M=6$:

$$\begin{aligned} u_r^i(k) &= A_i 1(k), \quad A_i = -0,75 + 0,25i, \quad i = \overline{1,5}, \\ u_r^6(k) &= \sin(2k\Delta_c), \quad t \in [0, \tau]. \end{aligned} \quad (4.27)$$

Длительность тренировочных входных сигналов выберем равной $\tau=10$ с.

Опишем параметры обучаемого НК вектором проектирования χ :

$$\chi = (w_{0,1}, w_{1,1}, w_{2,1}, w_{3,1}; \dots; w_{0,10}, w_{1,10}, w_{2,10}, w_{3,10}; w_0, w_1, \dots, w_{10}), \quad (4.28)$$

и определим следующую область поиска решения X :

$$\begin{aligned} w_{ij} &\in [-10, 10], \quad i = \overline{0,3}, \quad j = \overline{1,10} \\ w_i &\in [-10, 10], \quad i = \overline{0,10}. \end{aligned} \quad (4.29)$$

Длину каждой хромосомы примем равной $\lambda_i=14$, $i = \overline{1,51}$.

После 10000 эпох эволюции популяции из 100 особей с помощью ГА была получена оптимальная настройка НК. В табл. 4.8, 4.9 приведены его параметры.

Таблица 4.8

Параметры нейроконтроллера. Скрытый слой

Номер нейрона	w_{0i}	w_{1i}	w_{2i}	w_{3i}
1	-7,6329	-1,9880	-6,4170	9,1198
2	8,6352	0,3241	4,7970	-0,0922
3	-6,2498	0,2765	-5,2377	-6,5257
4	8,8451	-9,4409	1,1298	3,8790
5	-9,9036	-5,4502	-4,6493	-6,2840
6	-8,9538	-1,9490	1,0395	-7,4608
7	6,5257	-7,2655	4,3478	-8,6364
8	-0,0043	-5,0082	9,1430	-4,7250
9	-2,2005	2,4641	-1,5144	-2,5056
10	0,0153	-2,4995	-1,5608	6,2376

Таблица 4.9

Параметры нейроконтроллера. Выходной нейрон

Параметр	Значение	Параметр	Значение	Параметр	Значение
w_0	0,4547	w_4	-0,4920	w_8	-0,5626
w_1	-0,4762	w_5	0,3427	w_9	-0,1820
w_2	0,2257	w_6	0,3801	w_{10}	0,2511
w_3	-0,2251	w_7	-0,3780	—	—

На рис. 4.10 приведены переходные процессы в НСУ с объектом, моделируемым НЭ. Полученный НК обеспечивает такое же качество управления, как и синтезированный в подразделе 3.4: перерегулирование 15 %, статическая ошибка менее 0,005 при полутора колебаниях переходного процесса. Обученный НК имеет достаточный уровень обобщения как по амплитуде входного сигнала, так и по его частоте.

Как было сказано выше, вопрос в том, сможет ли НК с таким же успехом управлять самим объектом (то есть его дифференциальной моделью).

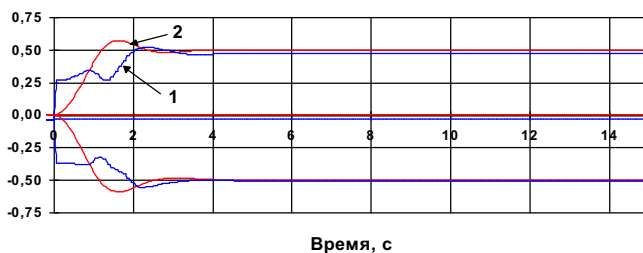


Рис. 4.10

Переходные процессы в нейроконтроллере и нейроэмуляторе объекта управления.

Кривая 1 — выход нейроконтроллера;
кривая 2 — реакция нейроэмулятора.

Входной сигнал задания $u_r=A1(t)$, $A=-0,5;0;0,5$

На рис. 4.11 приведены переходные процессы в НСУ реальным объектом (то есть моделируемым с помощью дифференциальных уравнений).

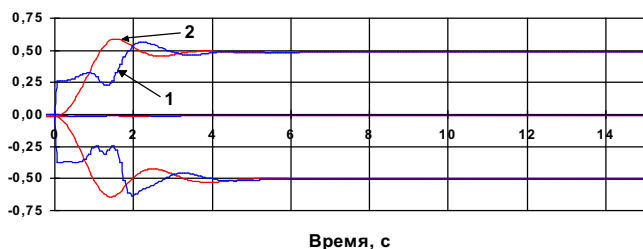


Рис. 4.11

Переходные процессы в НСУ.

Кривая 1 — выход нейроконтроллера;
кривая 2 — выход тестового динамического объекта.

Входной сигнал задания $u_r=A1(t)$, $A=-0,5;0;0,5$

Как видно, синтезированный НК обеспечивает такое же качество управления, что и для НЭ. Можно отметить только небольшое увеличение перерегулирования при сохранении остальных характеристик. При-

чиной этого является то, что во время обучения скорость изменения сигнала управления, генерируемого НК, все же превышала максимально допустимую для НЭ.

В целом, можно заключить, что приведенный пример доказывает возможность построения НСУ по нейросетевым же моделям объектов управления. Точность НЭ позволяет использовать их не только как «банки данных» о динамическом объекте, но и активно экспериментировать с ним, не теряя связи с реальностью. Это лишний раз говорит о парадигмоидальности нейроуправления в структуре общей теории управления [92].

Как и ранее, необходимо отметить, что линейность тестового объекта никак не уменьшает значимость полученных результатов. Синтез НСУ от начала до конца проводился без учета этой особенности объекта, и методика этого синтеза является общей для всех линейных и нелинейных объектов.

Время обучения НК составило около 8 часов вычислений на компьютере класса Pentium-100. Принимая во внимание, что НЭ был синтезирован за 3 часа вычислений, можно оценить общее время проектирования такой НСУ в 20 часов. Таким образом, при минимальном участии человека менее чем за сутки, используя только методы нейроуправления, можно построить СУ, готовую к аппаратной реализации. Это позволяет уже сегодня надеяться на возможность скорого создания большого количества промышленных систем искусственного интеллекта, решающих задачи управления [91].

4.3. Анализ параметрической и структурной чувствительности нейросетевой системы управления

В настоящее время во всем мире большое внимание уделяется цифровым СУ, реализованным на микропроцессорных устройствах, так как они обеспечивают большую гибкость при низкой себестоимости. По этим же причинам все существующие в виде электронных приборов ИНС были созданы на базе специализированных процессоров [52]. Однако подавляющее большинство работающих промышленных СУ все еще являются аналоговыми. Это особенно характерно для производств, связанных с тяжелыми внешними условиями функционирования СУ, в частности для энергетики. Поэтому НК, реализованные на аналоговых устройствах и работающие напрямую с аналоговыми сигналами, несмотря на кажущуюся «бесперспективность», могли бы уже сейчас использоваться в действующих технологических комплексах. При этом затраты на модернизацию были бы намного меньше, чем при внедрении цифровых НСУ.

Чтобы программно синтезированная НСУ могла быть реализована аппаратно, она должна отвечать ряду технических требований. Определяющим среди них являются стабильность ее статических и динамических характеристик при изменениях расчетных параметров НК. Такие изменения практически всегда присутствуют в реальных электронных устройствах из-за неточностей изготовления, ошибок настройки и ухода параметров полупроводниковых элементов под действием факторов внешней среды и времени.

Чтобы выяснить, каковы пределы нечувствительности НСУ, была выбрана НСУ тестовым динамическим объектом, синтезированная в подразделе 3.4 (см. рис. 3.12). В табл. 3.5, 3.6 приведены значения параметров обученного с помощью ГА НК (3.22) на базе 3-5-1 МСП. На

рис. 3.13 представлены результаты моделирования работы НСУ при различных тестовых входных воздействиях.

Для оценки параметрической чувствительности НСУ введем интегральный функционал качества I вида:

$$I(y) = \frac{1}{M_e} \sum_{i=1}^{M_e} \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} (y_i(k) - u_r^i(k))^2. \quad (4.30)$$

где $M_e=3$ — количество экспериментов,

$\tau=10$ с — время анализа переходного процесса в системе.

Дискретность данной НСУ равна $\Delta=0,1$ с.

В качестве тренировочных входных воздействий на НСУ u_r^i выберем скачкообразные сигналы:

$$u_r^i(k) = A_i 1(k),$$

$$A_i = -2 + i, \quad i = \overline{1,3}. \quad (4.31)$$

В этом случае (4.30) характеризует в основном статические характеристики НСУ. Минимальное значение функционала качества I , соответствующее номинальной настройке НК, равно $I=0,0235$.

Для упрощения ссылки на параметры НК пронумеруем их от 1 до 26. В табл. 4.10 приведены номера и условные обозначения синаптических весов и сдвигов всех нейронов скрытого и выходного слоев.

Таблица 4.10

Нумерация параметров нейроконтроллера

Номер параметра	Параметр	Номер параметра	Параметр	Номер параметра	Параметр
1	$w_{0,1}$	10	$w_{1,3}$	19	$w_{2,5}$
2	$w_{1,1}$	11	$w_{2,3}$	20	$w_{3,5}$
3	$w_{2,1}$	12	$w_{3,3}$	21	w_0
4	$w_{3,1}$	13	$w_{0,4}$	22	w_1
5	$w_{0,2}$	14	$w_{1,4}$	23	w_2

6	$w_{1,2}$	15	$w_{2,4}$	24	w_3
7	$w_{2,2}$	16	$w_{3,4}$	25	w_4
8	$w_{3,2}$	17	$w_{0,5}$	26	w_5
9	$w_{0,3}$	18	$w_{1,5}$	—	—

Также введем вектор параметров НК \mathbf{w} :

$$\mathbf{w} = (w^1, w^2, \dots, w^{26}), \quad (4.32)$$

верхний индекс его координат является номером параметра. Вектор номинальных значений параметров НК обозначим \mathbf{w}_0 .

Для того чтобы оценить параметрическую чувствительность НСУ, проварьируем каждый из 26-и параметров НК в пределах $-100 \div 100$ % от его номинального значения:

$$w^i = (1 + \beta)w_0^i, \quad \beta \in [-1, 1], \quad i = \overline{1, 26} \quad (4.33)$$

На рис. 4.12 представлен график изменения функционала качества в зависимости от значения параметров. Как видно, варьирование значения любого из 26-и параметров НК в пределах $-20 \div 20$ % от номинального не приводит более, чем к двукратному увеличению функционала I . При этом качество работы НСУ ухудшается незначительно. Возникающая в НСУ статическая ошибка не превышает 5 %, перерегулирование остается на уровне 15 % при 1,5—2 колебаниях переходного процесса. Наличие такого коридора объясняется, прежде всего, распределенностью процесса вычисления между отдельными нейронами по всей структуре ИНС.

Стабильность характеристик стандартных полупроводниковых приборов, применяемых в промышленных системах автоматического управления, намного превышает 20 %. Благодаря этому построенный на их базе НК сможет сохранять свои свойства во всех режимах работы, а

значит, сегодня вполне реально говорить об аппаратной реализации и внедрении в производство НСУ.

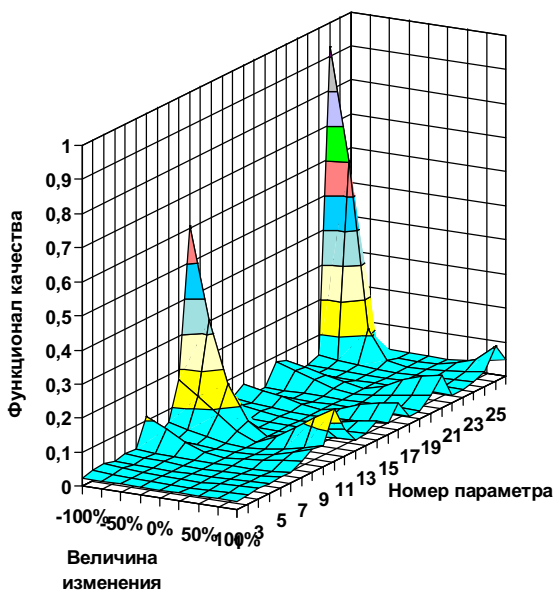


Рис. 4.12

Зависимость качества работы НСУ от изменения параметров нейроконтроллера

Из рис. 4.12 видно также, что разные параметры ИНС оказывают различное влияние на качество работы НСУ. Так изменения параметров w^{10} , w^{11} и w^{24} (веса первой и второй связей третьего нейрона скрытого слоя и вес связи с ним выходного нейрона, соответственно) вызывают наибольшие нарушения в работе НК. В то же время даже 100 %-ые изменения параметров w^4 , w^{17} и w^{22} (вес третьей связи первого и смещение пятого нейронов скрытого слоя и вес первой связи выходного нейрона, соответственно) практически не сказываются на эффективности управления. Это говорит о том, что различные нейроны делают разный вклад в преобразование вход-выход, реализуемое ИНС.

На рис. 4.13,а представлено семейство переходных функций НСУ при изменении значения 11-го параметра НК w^{11} в диапазоне от -100% до 100% . Изменения мене 50% от номинального значения оказывают влияние в основном на статическую ошибку работы НСУ, большие изменения нарушают также и ее динамические свойства, увеличивая колебательность переходной характеристики НСУ. Семейство, представленное на рис. 4.13,б, показывает, что изменение менее чувствительного 17-го параметра w^{17} незначительно сказывается лишь на статической ошибке системы.

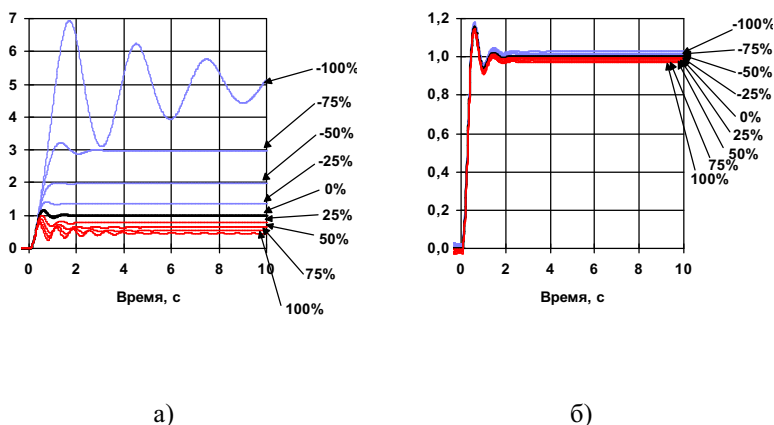


Рис. 4.13

Переходные функции системы управления при различных изменениях параметров нейроконтроллера.

- а) — переходные функции при изменении 11-го параметра и
- б) — при изменении 17-го параметра нейроконтроллера

Кроме стабильности характеристик НСУ при изменении параметров НК, важным ее качеством является безопасность работы в условиях различных аварий. Рассмотрим, как ведет себя НСУ при повреждении структуры НК в виде обрыва межнейронных связей или выхода из строя нейронов.

На рис. 4.14 представлен график изменения функционала качества работы НСУ при последовательном, в произвольном порядке, выходе из строя параметров НК (сигналы от поврежденных связей считаются нулевыми $w^j=0$). Как видно, благодаря тому, что функции, выполняемые ИНС, распределены между отдельными нейронами, при авариях в НК наблюдается не резкое пропадание, а постепенная *деградация* его работоспособности [93].



Рис. 4.14

Изменение качества работы системы управления при последовательном выходе из строя связей нейроконтроллера.

Кривая 1 — функционал качества для поврежденной системы;
кривая 2 — для системы без повреждений

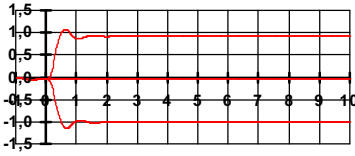
После выхода из строя более трети всех связей в НК, качество работы НСУ, все еще остается приемлемым. Резкое ухудшение наступает тогда, когда поврежденными оказываются наиболее критические связи (те же, что и обладают наибольшей чувствительностью к изменению их параметров). Однако, даже в этом случае, НСУ не оказывает разрушительного действия на объект, так как величина функционала качества остается ограниченной. При дальнейшем увеличении количества повреждений в НК, вплоть до его полного разрушения, сигнал управления становится нулевым, тем самым НСУ прекращает свою работу и «выключает» объект.

Это важное свойство ИНС позволяет использовать НСУ для автоматизации ответственных и опасных процессов, увеличивая их безопасность и живучесть.

Как было уже выяснено ранее, различные нейроны скрытого слоя играют в НК разные роли. Анализируя переходные процессы в системе при их отключении (на выходе нейрона нулевой сигнал) можно выяснить распределение функций между ними. Рис. 4.15 показывает, что ключевая роль в данном МСП принадлежит третьему нейрону скрытого слоя. При его аварии (см. рис. 4.15,в) сеть теряет способность различать полярность и амплитуду входного сигнала задания u_r . Как следствие, генерируемый на выходе НК сигнал управления и переходной процесс в НСУ является одинаковыми для всех u_r .

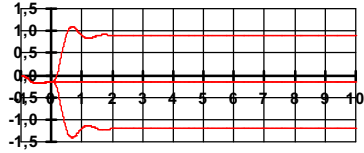
Четвертый и пятый скрытые нейроны определяют, прежде всего, динамические свойства НК, так как при их отключении (см. рис. 4.15,г,д) резко увеличивается колебательность системы. Роль первого и второго нейронов, очевидно, вспомогательная, потому что их выход из строя заметен только при определенных величинах входного сигнала u_r (см. рис. 4.15,а,б).

Такое распределение функция между отдельными нейронами НК не является обязательным. Ни количество нейронов скрытого слоя, ни их нумерация не позволяют определить роль, выполняемую тем или иным нейроном. В то же время, анализ других вариантов настройки НК, обеспечивающих такое же качество управления, показывает, что в общем, скрытые нейроны делятся на две группы. Первая из них определяет динамические свойства переходной характеристики НСУ, вторая — статические. Соотношение размеров каждой группы зависит только от сложности решаемой задачи управления.



Время, с

а)



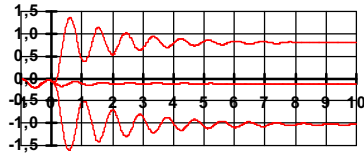
Время, с

б)



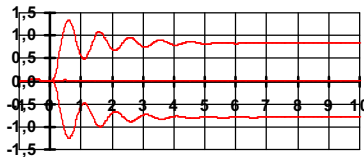
Время, с

в)



Время, с

г)



Время, с

д)

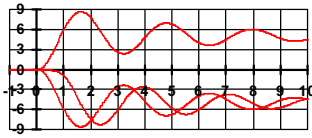
Рис. 4.15

Переходные процессы в НСУ при повреждении
скрытых нейронов нейроконтроллера.

Сигнал задания $u_r = A1(k)$; $A = -1; 0; 1$

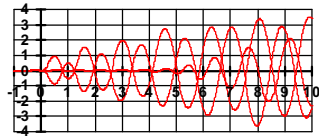
а) — поврежден 1-й нейрон; б) — 2-й; в) — 3-й; г) — 4-й; д) — 5-й

Одной из аварий, часто возникающих при практическом использовании СУ, является обрыв обратных связей по выходу объекта управления. Причиной этого может быть либо выход из строя датчиков, либо повреждение в линиях связи. На рис. 4.16 представлены переходные процессы при различных вариантах такой аварии, при этом, как и ранее, считается, что сигнал в поврежденных линиях нулевые.



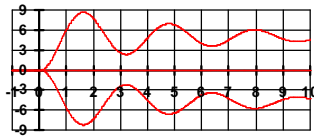
Время, с

а)



Время, с

б)



Время, с

в)

Рис. 4.16

Переходные процессы в НСУ при обрыве обратных связей.

Сигнал задания $u_r = A1(k)$; $A = -1; 0; 1$.

а) — повреждена обратная связь по y_k ;

б) — по y_{k-1} ; в) — по y_k и y_{k-1}

Как показывают переходные процессы в НСУ (см. рис. 4.16,а), в случае обрыва обратной связи по выходному сигналу y_k резко ухудшается статическая характеристика системы. Динамические свойства НСУ также изменяются, приближаясь к свойствам самого объекта управления. В результате аварии поврежденная НСУ и объект управления имеют примерно одинаковые постоянную времени, затухание и перерегулирование.

Обрыв обратной связи по задержанному выходному сигналу y_{k-1} (см. рис. 4.16,б) вызывает в НСУ колебания с частотой в три раза превышающей собственную частоту объекта управления. Возникающий

колебательный процесс является устойчивым, в установившемся режиме его амплитуда остается постоянной и ограниченной.

В случае аварии в обоих каналах обратной связи по y_k и y_{k-1} (см. рис. 4.16,в) НК превращается в линейный усилитель с коэффициентом усиления 3,8. Как показывают эксперименты с другими вариантами настройки НК для такого объекта управления, эта ситуация является общей. Коэффициент усиления НСУ может меняться, но для сигнала u , НК является линейным усилителем.

Такая линейность не является внутренним свойством ИНС, а достигается в процессе тренировки. Объясняется она тем что цель обучения НК — получение линейной следящей НСУ. Так как сам тестовый объект управления линеен, то естественно, и контроллер должен быть линеен по задающему воздействию.

Как показывают аналогичные исследования других НСУ, полученные в этом подразделе результаты анализа параметрической и структурной чувствительности тестовой НСУ можно считать общими для всего класса НСУ. Следовательно, уже сегодня НСУ могут использоваться в промышленных условиях в виде универсальных микропроцессорных комплексов, математически моделирующих работу ИНС, или специальных устройств на базе аналоговых моделей отдельных нейронов.

4.4. Выводы

Результаты выполненных исследований доказывают, что предложенная на базе ГА методика стимулируемого обучения НК позволяет синтезировать НСУ как линейными, так и нелинейными динамическими объектами. При этом необходимый объем вычислений и требуемая структура НК зависят не столько от степени нелинейности, сколько от

порядка динамического объекта управления. Для большинства практических задач управления верхнюю границу сложности НК можно оценить как 20—25 входных и 50—60 скрытых нейронов.

Приведенный пример синтеза НСУ по НЭ динамического объекта доказывает, что задачи управления в их общей постановке возможно решать, используя на всех этапах только методы эволюционного моделирования. Корректность работы НК с дифференциальной моделью объекта также еще раз подтверждает высокую точность нейросетевой идентификационной модели.

Важным выводом, является также то, что эффективность получаемого с помощью предложенной методики решения во многом определяется видом функционала качества. Выбор функционала, используемого для оценки работы синтезируемой НСУ, как и структуры самого закона управления, является задачей, решаемой исследователем в зависимости от сложности рассматриваемого объекта управления. Можно определить основные приемы, позволяющие облегчить процесс поиска оптимальной настройки НК. Это — повышение степени, в которую возводится ошибка управления, введение времени в качестве множителя при этой ошибке или формирование функционала качества в виде свертки сигналов ошибки и ее производной.

Анализ эффективности работы НК при различных параметрических изменениях, а также в присутствии шумов в канале обратной связи доказал возможность его аппаратной реализации и использования в реальных СУ. Положительным свойством НСУ является низкая чувствительность к уходу параметров, что значительно упрощает настройку НК и повышает устойчивость работы в условиях меняющейся внешней среды. Существенная распределенность вычислительного процесса в ИНС, обеспечивает повышенную, по сравнению с традиционными контроллерами, безопасность СУ, так как качество работы НСУ ухудшается с рос-

том количества повреждений постепенно. Выход нейронов НК из строя или обрыв обратных связей также не приводят к мгновенным разрушительным процессам.

Исходя из полученных результатов, следует заключить, что уже сегодня практически возможно создавать разнообразные промышленные НСУ, пригодные для использования в действующих технологических комплексах.

ЗАКЛЮЧЕНИЕ

Искусственные нейронные сети представляют собой мощное средство решения задач управления различного уровня сложности. Методы контролируемого обучения нейронных сетей позволяют синтезировать по экспериментальным данным модели сложных нелинейных объектов управления, а при осознании целей управления — нейроконтроллеры, обеспечивающие оптимальное поведение объекта.

Главным научно-практическим итогом диссертации являются методики эволюционного синтеза нейросетевых идентификационных моделей и нейроконтроллеров на базе трехслойных прямонаправленных искусственных нейронных сетей. Отличительной чертой методик является оригинальная форма кодирования параметров скрытого слоя многослойного перцептрона, а также наличие процедуры коррекции структуры скрытого слоя РБФ-сети.

Результативность разработанных методик продемонстрирована на примере синтеза нейросетевой системы управления для линейного и нелинейного тестовых динамических объектов, в том числе с заданными частотными характеристиками, а также на примере построения нейросетевой идентификационной модели тестового объекта при наличии шумов в экспериментальных данных.

Существенно, что ни структура нейроконтроллера, ни время его обучения в рамках разработанной методики не увеличиваются катастрофически с возрастанием сложности объекта управления.

Для большинства практических задач верхнюю границу сложности нейроконтроллера можно оценить как 20—25 нейронов входного слоя и 50—60 нейронов скрытого слоя. А это значит, что для синтеза соответствующей нейросетевой системы управления вполне достаточно персонального компьютера и 50—100 часов машинного времени.

В работе показано, что скачкообразные тренировочные сигналы гарантируют синтез НСУ с малой статической ошибкой, однако переходная характеристика НСУ, обучаемой только по сигналам вида $u_r=A1(t)$, может быть сильно колебательной. Для того чтобы обеспечить НСУ необходимые динамические характеристики, в тренировочные воздействия необходимо включать гармонические сигналы или модифицировать функционал качества, согласно которому ведется обучение нейроконтроллера.

В работе выполнено сравнение особенностей построения нейроконтроллеров на базе традиционного ПИД-закона управления и в рамках нейрокибернетического подхода. Результаты показывают, что главным преимуществом НСУ на базе ПИД-нейроконтроллера является легкая интегрируемость в действующие непрерывные системы управления, в то время как дискретные НСУ отличаются большей универсальностью и шириной круга решаемых задач. Оба типа нейроконтроллеров демонстрируют характеристики на уровне лучших образцов, синтезированных в рамках классической теории управления.

Генетический алгоритм глобальной оптимизации выступает единой платформой, на которой объединяются методы обучения нейроэмуляторов и нейроконтроллеров. Выполненный в работе синтез нейросетевой системы управления с использованием нейроэмулятора динамического объекта доказывает, что, используя только методы эволюционного моделирования, можно решать любые задачи управления. Тем самым, с одной стороны, подтверждается парадигмоидальность нейроуправления, и с другой стороны, демонстрируются уникальные свойства симбиоза генетических алгоритмов и искусственных нейронных сетей. Именно совместное использование двух этих методов эволюционного моделирования может обеспечить дальнейшее успешное развитие нейрокибернетики.

Наиболее перспективной областью внедрения результатов диссертации является, по-видимому, энергетика, которая характеризуется относительно низкой скоростью протекания нормальных рабочих режимов, многогранностью факторов, влияющих на характеристики объектов, неоднозначностью выбора критериев качества функционирования.

Впрочем, бурно протекающий процесс информатизации многих прикладных технических областей существенно расширяет круг потенциальных реципиентов нейрокибернетических вычислительных технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ramon y Cajal S. Histologie du systeme nerveux de l'homme et des vertebres. — Paris: Maloine, 1911. — 714 p.
2. Shepherd G. M., Koch C. Introduction to synaptic circuits // The Synaptic Organization of the Brain (G. M. Shepherd, ed.). — New York: Oxford University Press, 1990. — P. 3—31.
3. Churchland P. S. Neurophilosophy: Toward a Unified Science of the Mind/Brain. — Cambridge, MA: MIT Press, 1986. — 127 p.
4. Aleksander I., Morton H. An Introduction to Neural Computing. — London: Chapman & Hall, 1990. — 218 p.
5. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics. — 1943. — N 5. — P. 115—133.
6. Винер Н. Кибернетика или управление и связь в животном и машине. — М.: Сов. радио, 1968. — 357 с.
7. Hebb D. O. The Organization of Behavior: A Neuropsychological Theory. — New York: Wiley, 1949. — 358 p.
8. Rochester N., Holland J. H., Haibt L. H., Duda W. L. Tests on a cell assembly theory of the action of the brain, using a large digital computer. // IRE Transactions on Information Theory. — 1956. — N IT-2. — P. 80-93.
9. Uttley A. M. A theory of the mechanism of learning based on conditional probabilities // Proc. of the 1st International Conference on Cybernetics. — Namur, Gauthier-Villars, Paris (France). — 1956. — P. 83—92.
10. Uttley A. M. Information Transmission in the Nervous System.— London: Academic Press, 1979. — 215 p.
11. Ashby W. R. Design for a Brain. — New York: Wiley, 1952. — 306 p.

12. Minsky M. L. Theory of neural-analog reinforcement systems and its application to the brain-model problem: Ph.D. Thesis. — Princeton University, Princeton, NJ., 1954. —143 p.
13. Minsky M. L. Steps toward artificial intelligence // Proceedings of the Institute of Radio Engineers. — 1961. — N 49. — P. 8—30.
14. Winograd S., Cowan J. D. Reliable Computation in the Presence of Noise. — Cambridge, MA: MIT Press, 1963. —247 p.
15. Rosenblatt F. The Perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review.— 1958. — N 65. — P. 386—408.
16. Widrow B., Hoff M.E., Jr. Adaptive switching circuits // IRE WES-CON Convention Record. — 1960. — P. 96—104.
17. Widrow B. Generalisation and information storage in networks of adaline «neurons» // Self-Organizing Systems (M.C. Yovitz, G.T. Jacobi, and G.D. Goldstein, eds.). — Washington, D.C.: Sparta, 1962. — P. 435—461.
18. Cowan J. D. A Mathematical Theory of Central Nervous Activity: Ph. D. Thesis. — University of London, UK, 1967. —178 p.
19. Минский М., Пейперт С. Перцептроны. — М.: Мир, 1971. —261 с.
20. Willshaw D. J., von der Malsburg C. How patterned neural connections can be set up by self-organization // Proceedings of the Royal Society of London. — 1976. — Series B. — N 194. — P. 431—445.
21. Kohonen T. Self-organized formation of topologically correct feature maps. // Biological Cybernetics. — 1982.— N 43. — P. 59—69.
22. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors // Nature (London). — 1986. — N 323. — P. 533—536.
23. Learning-logic: Casting the cortex of the human brain in silicon, Parker D. B.: Technical Report / Center for Computational Research in Eco-

nomics and Management Science. — TR-47. — Cambridge, MA: MIT, 1985. — 73 p.

24. LeCun, Y. Une procedure d'apprentissage pour reseau a seuil asymetrique // *Cognitiva* 85. — 1985. — P. 599—604.

25. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral science: Ph.D. Thesis. — Harvard University, Cambridge, MA, 1974. —120 p.

26. Broomhead D. S., Lowe D. Multivariable functional interpolation and adaptive networks // *Complex Systems*. — 1988. — N 2. — P. 321—355.

27. Bashkirov O. A., Bravermann E. M., Muchnik I. B. Potential function algorithms for pattern recognition learning machines // *Automation and Remote Control*. — 1964. — N 25. — P. 629—631.

28. On System Identification Using Pulse-Frequency Modulated Signal. Bondarev V. N.: EUT Report / Eindhoven University of Technology, Netherlands. — ISBN 90-6144-195-1. —88-E-195. — Eindhoven, 1988. — 84 p.

29. Mendel J. M., McLaren R. W. Reinforcement-learning control and pattern recognition systems. // *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications* (J. M. Mendel and K. S. Fu, eds.). — New York: Academic Press, 1970. — P. 287—318.

30. Haykin S. *Neural Networks. A comprehensive foundation*. — New York, NY: Macmillan, 1994. —696 p.

31. Sutton R. S., Barto A. G., Williams R. J. Reinforcement learning is direct adaptive optimal control // *Proc. of the American Control Conference*. — Boston, MA. — 1991. — P. 2143—2146.

32. Holland J. H. *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence*. — London: Bradford book edition, 1994 — 211 p.

33. Schaffer J. D., Whitley D., Eshelman L. J. Combination of Genetic Algorithms and Neural Networks: A Survey of the State of the Art // *Proc.*

Int. Workshop on Combinations of Genetic Algorithms and Neural Networks (Eds. L. D. Whitley, J. D. Schaffer).— Baltimore, Maryland. — 1992.— P. 1—3.

34. Poggio T., Girosi Networks for approximation and learning // Proceedings of the IEEE. — 1990. — N 78. — P. 1481—1497.

35. Cybenko G. Approximation by Superpositions of Sigmoidal Function. — Urbana: University of Illinois, 1988. — 280 p.

36. Funahashi K. On the approximate realization of continuous mappings by neural networks. // Neural Networks. — 1989.— N 2. — P. 183—192.

37. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // Neural Networks. — 1989. — Vol. 2. — P. 359—366.

38. Barron A. R. Neural net approximation // Proc. of the Seventh Yale Workshop on Adaptive and Learning Systems. — New Haven, CT: Yale University. — 1991. — P. 69—72.

39. Bolt G. R. Fault Tolerance in Artificial Neural Networks: D. Phil. Thesis. — York University, Ontario, 1992. —215 p.

40. Неймарк Ю. И., Коган Н. Я., Савельев В. П. Динамические модели теории управления. — М.: Наука, Главная редакция физико-математической литературы, 1985. —400 с.

41. Werbos P. J. Backpropagation and neurocontrol: A review and prospectus // Proc. of International Joint Conf. On Neural Networks. — Vol. 1. — Washington, DC. — 1989. — P. 209—216.

42. Цыпкин Я. З. Адаптация и обучение в автоматических системах. — М.: Наука, 1968. — 400 с.

43. Jones A. H. Genetic Tuning Of Non-Linear PID Controllers // Artificial Neural Nets and Genetic Algorithms. Proceedings of the International

- Conference in Ales, France, 1995 (D. W. Pearson, N. C. Steele, R. F. Albrecht, eds.). — Wien: Springer Verlag, 1995. — P. 412—415.
44. Nguyen D. H., Widrow B. Neural networks for self-learning control systems // IEEE Control Systems Magazine. — 1990. — Vol. 10. — No. 3. — P. 334—341.
45. Шульце К.-П., Реберг К.-Ю. Инженерный анализ адаптивных систем: Пер. с нем. — М.: Мир, 1992. — 280 с.
46. Kuntanapreeda S., Gunderson R. W., Fullmer R. R. Neural Network Model Reference Control of Nonlinear Systems // Proc. Of Int. Joint Conf. on Neural Networks. — Vol. II. — Baltimore, Maryland. — 1992. — P. 94—99.
47. Narendra K. S., Annaswamy A. M. Stable Adaptive Systems. — Englewood Cliffs, NJ: Prentice-Hall, 1989. — 236 p.
48. Narendra K.S., Parthasarathy K. Identification and Control of Dynamical Systems Using Neural Networks // IEEE Transactions on Neural Networks. — 1990. — Vol. 1. — No. 1. — P. 4—27.
49. Palis F., Schmied Th., Skljarenko E. A. Fuzzy and Neurocontrol of Drive Systems with changing Parameters and Load // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1995, С. 116—119.
50. Kuan A., Bavarian B. Compensation of Unmodeled Friction in Manipulators using Neural Networks // Proc. Of Int. Joint Conf. on Neural Network. — Vol. II. — Baltimore, Maryland. — 1992. — P. 817—822.
51. Park J, Sandberg I. W. Universal approximation using radial-basis-function networks // Neural Computations. — 1991. — N 3. — P. 246—257.
52. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. — Х.: Основа, 1997. — 112 с.

53. Sergeev S. A., Makhotilo K. V. Evolutionary Synthesis of Dynamical Object Emulator Based on RBF Neural Network // Proc. of the First Online Workshop on Soft Computing WSC1. — Nagoya: Nagoya University. — 1996. — P. 31—36.

54. Zhang B.-T., Muhlenbein H. Evolving optimal neural networks using genetic algorithms with Occam's razor // Complex systems. — 1993. — N 7(3). — P. 199—220.

55. Goldberg D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. — Addison-Wesley, 1989. — 412 p.

56. Goldberg D. E., Smith R. E. Nonstationary function optimization using genetic algorithms with dominance and diploidy // Proc. of Second Int. Conf. on Genetic Algorithms and Their Applications. — Cambridge, MA: Lawrence Erlbaum.— 1987. — P. 59—68.

57. Klepikov V. B., Sergeev S. A., Mahotilo K. V., Lysenko L. I. Diploidy-based Genetic Algorithm in Non-stationary Environment // Труды конференції с міжнародним участієм «Проблеми автоматизованого електропривода». — Харьков: Основа. — 1995. — С. 108.

58. Сергеев С. А., Махотіло К. В. Методологічні основи еволюційного моделювання технічних пристроїв як диплоїдних організмів // Тези 1-ї Міжнародної науково-технічної конференції «Математичне моделювання в електротехніці й електроенергетиці». — Львів. — 1995. — С. 39.

59. Klepikov V. B., Sergeev S. A., Mahotilo K. V. A genetic algorithm based on diploid representation of an individual: pro-and-con study // Proc. of XXII International school and conference on Computer Aided Design. — Gurzuff (Ukraine). — 1995. — P. 42—43.

60. Химмельблау Д. Прикладное нелинейное программирование. — М.: Мир, 1975. — 534 с.

61. Ивахненко А. Г. Принятие решений на основе самоорганизации. — М.: Сов. радио, 1976. — 363 с.

62. Махотило К. В., Бикой Ж. А., Малафеев Н. П. Исследование влияния численности популяции на эффективность эволюционного поиска // Труды Междунар. научно-технической конференции «microCAD'95». Информационные технологии: наука, техника, технология, образование, здоровье» — Часть 2. — Харьков. — 1995. — С. 35.

63. Klepikov V. B., Sergeev S. A., Mahotilo K. V. Modification of Holland's reproductive plan for diploid populations // Artificial Neural Nets and Genetic Algorithms. Proceedings of the International Conference in Ales, France, 1995 (D. W. Pearson, N. C. Steele, R. F. Albrecht, eds.). — Wien: Springer Verlag, 1995. — P. 337—339.

64. Сергеев С. А., Махотило К. В., Бикой Ж. А. Эволюционные алгоритмы в оптимальном проектировании электромеханических систем // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков. — 1994. — С. 84—86.

65. Сергеев С. А., Махотило К. В. Генетические алгоритмы в синтезе прямонаправленных нейронных сетей // Труды XXIII международной конференции и дискуссионного научного клуба «Новые информационные технологии в науке, образовании и бизнесе». — Ялта—Гурзуф, Крым. — 1996. — С. 338—342.

66. Петрашев С. Н. Метод синтеза нейросетевых моделей динамических объектов // Труды Международной научно-технической конференции «microCAD'97». Информационные технологии: наука, техника, технология, образование, здоровье. — Часть 3. — Харьков. — 1997. — С. 76—80.

67. Петрашев С. М. Розроблення і впровадження нейросіткових технологій математичного моделювання для розв'язування прикладних

проблем теплоенергетики: Автореф. дис... канд. техн. наук: 05.13.07 / Одеський держ. політехнічний ун-т. — Одеса, 1997. — 18 с.

68. Вороновский Г. К., Махотило К. В., Петрашев С. Н. Эволюционный синтез предиктора динамического объекта управления на базе RBF-нейронных сетей // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1996. — С. 276—279.

69. Махотило К. В., Бутуфену Нейросетевой метод идентификации динамических объектов // Труды Международной научно-технической конференции «microCAD'96». Информационные технологии: наука, техника, технология, образование, здоровье. — Часть 1. — Харьков. — 1996. — С. 127.

70. Махотило К. В. GENNETICS: пакет прикладних програм для розв'язання оптимізаційних задач // Тези 1-ї Міжнародної науково-технічної конференції «Математичне моделювання в електротехніці й електроенергетиці». — Львів. — 1995. — С. 202.

71. Вороновский Г. К., Махотило К. В., Сергеев С. А., Сергеенкова Г. Г. Краткосрочное предсказание электропотребления для крупного жилого массива города // Proc. of Third International Scientific and Technical Conference on Unconventional Electromechanical and Electrical Systems «UEES'97». — Alushta, The Crimea (Ukraine). — 1997. — P. 1073—1078.

72. Вороновский Г. К., Махотило К. В., Сергеев С. А. Предсказание электрической нагрузки при помощи искусственных нейронных сетей // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1997. — С. 369—370.

73. Махотило К. В., Сергеев С. А., Сушков А. В. Нейросетевая идентификация динамических объектов по зашумленным данным //

Вестник Харьковского государственного политехнического университета. — 1998. — Выпуск 11. — С. 3—7.

74. Сергеев С. А., Махотило К. В. Одношаговые и краткосрочные предикторы: проблема точности настройки и глубины предсказания // Праці 4-ї Української конференції з автоматичного управління «Автоматика-97». — Том 2. — Черкаси. — 1997. — С. 108.

75. Klepikov V. B., Sergeev S. A., Mahotilo K. V., Voronovsky G. K. Neural Technologies in Electrical Drive Control // Proc. of Conf. «Sterowanie w Energoelektronice i Napędzie Elektrycznym». — Lodz—Arturovek (Poland). — 1995. — P. 336—343.

76. Bertrand D. J., Collins D. J. Neural Network Controllers for the X29 Aircraft // Proc. Of Int. Joint Conf. on Neural Network. — Vol. I. — Baltimore, Maryland. — 1992. — P. 321—326.

77. Kazmierkowski M. P., Sobczuk D. L., Dzieciakowski M. A. Artificial Neural Networks Based Robust Current Regulator For Three Phase PWM Inverters // Proc. of Conf. «Sterowanie w Energoelektronice i Napędzie Elektrycznym». — Lodz—Arturovek (Poland). — 1995. — P. 325—330.

78. Lansberry J. E., Wozniak L., Goldberg D. E. Optimal Hydrogenerator Governor Tuning with Genetic Algorithm // Transaction on Energy Conversion. — 1992. — V9. — N4. — P. 48—57.

79. Клепиков В. Б., Сергеев С. А., Махотило К. В. Следящий электропривод постоянного тока на базе искусственных нейронных сетей // Праці 3-ї Української конференції з автоматичного керування «Автоматика-96». — Том 3. — Севастополь. — 1996. — С. 28.

80. Клепиков В. Б., Сергеев С. А., Махотило К. В., Вороновский Г. К. Искусственные нейронные сети: новая парадигма в управлении // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1995. — С. 111—115.

81. Вороновский Г. К., Махотило К. В., Сергеев С. А. Синтез робастного ПИД-нейроконтроллера для регулирования температуры пара на выходе котла // Вестник Харьковского государственного политехнического университета. Экология. Математика. Электроэнергетика. — 1997. — Выпуск 8. — С. 121—125.
82. Арушанян О. Б., Залеткин С. Ф. Численное решение обыкновенных дифференциальных уравнений на Фортране. — М.: Московский университет, 1990. — 352 с.
83. Klerikov V. B., Sergeev S. A., Mahotilo K. V., Amplitude-Frequency Characteristic of a Neural Control Based DC Drive // Tagungsband «Leistungselektronische Aktoren und intelligente Bewegungssteuerungen». — Magdeburg (Deutschland). — 1996. — P. 47—53.
84. Troudet T., Garg S., Merrill W. Design and Evaluation of a Robust Dynamic Neurocontroller for a Multivariable Aircraft Control Problem// Proc. Of Int. Joint Conf. on Neural Network. — Vol. I. — Baltimore, Maryland. — 1992. — P. 308—314.
85. Махотило К. В. Электропривод постоянного тока с использованием искусственных нейронных сетей // Труды Международной научно-технической конференции «microCAD'96». Информационные технологии: наука, техника, технология, образование, здоровье». — Часть 1. — Харьков. — 1996. — С. 91.
86. Махотило К. В., Сергеев С. А., Сушков А. В. Дискретная нейросетевая система управления нелинейным динамическим объектом // Вестник Харьковского государственного политехнического университета. — 1998. — Выпуск 10. — С. 13—18.
87. Клепиков В. Б., Махотило К. В., Обруч И. В., Осичев А. В., Камаль Наджи Нейронное управление электромеханической системой с отрицательным вязким трением // Труды конференции с международ-

ным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1996. — С. 283—286.

88. Клепиков В. Б., Махотило К. В., Обруч И. В. Синтез нейросетевой системы управления одномассовой электромеханической системы с отрицательным вязким трением при ограничении координат электропривода // Труды конференции с международным участием «Проблемы автоматизированного электропривода». — Харьков: Основа. — 1997. — С. 19—21.

89. Мань Н. В. Оптимизация настройки робастных регуляторов с помощью «оврагоперешагового» алгоритма нелинейной минимизации // Теплоэнергетика. — 1995. — № 10. — С. 58—65.

90. Ribeiro B., Dourado A., Costa E. Industrial Kiln Multivariable Control: MNN And RBFNN Approaches // Artificial Neural Nets and Genetic Algorithms. Proceedings of the International Conference in Ales, France, 1995 (D. W. Pearson, N. C. Steele, R. F. Albrecht, eds.). — Wien: Springer Verlag, 1995. — P. 408—411.

91. Sergeev S. A., Mahotilo K. V., Voronovsky G. K., Petrashev S. N. ANN-based intelligent system for process control // Proc. of 1st Int. Conf. on Evolutionary Computation and Its Applications «EvCA'96». — Moscow (Russia). — 1996. — P. 330—334.

92. Klepikov V. B., Sergeev S. A., Mahotilo K. V., Voronovsky G. K. ANN+GA-based Intelligent Control System // Proc. of The 4th International Workshop on Optimimization and Inverse Problems in Electromagnetism. — Brno (Czech Republic). — 1996. — P. 51.

93. Махотило К. В. Анализ параметрической чувствительности нейросетевой системы управления // Труды Международной научно-технической конференции «microCAD'97». Информационные технологии: наука, техника, технология, образование, здоровье. — Часть 3. — Харьков. — 1997. — С. 137—141.