

## 7. Нейросетевые системы управления

### 7.1. Общие сведения о нейронных системах

В наши дни возрастает необходимость в системах, которые способны не только выполнять однажды запрограммированную последовательность действий над заранее определенными данными, но и способны сами анализировать вновь поступающую информацию, находить в ней закономерности, производить прогнозирование и т.д. В этой области приложений самым лучшим образом зарекомендовали себя так называемые *нейронные сети* – самообучающиеся системы, имитирующие деятельность человеческого мозга. Рассмотрим подробнее структуру искусственных нейронных сетей (НС) и их применение в конкретных задачах.

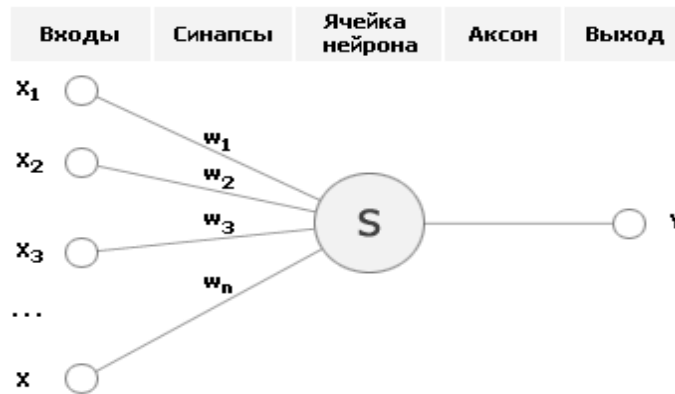


Рис. 1. Схема нейрона

Из рис 17.1 видно, что искусственный нейрон, так же, как и живой, состоит из синапсов, связывающих входы нейрона с ядром; ядра нейрона, которое осуществляет обработку входных сигналов и аксона, который связывает нейрон с нейронами следующего слоя. Каждый синапс имеет вес, который определяет, насколько соответствующий вход нейрона влияет на его состояние. Состояние нейрона определяется по формуле

$$S = \sum_{i=1}^n x_i w_i \quad (7.1)$$

где

$n$  – число входов нейрона

$x_i$  – значение  $i$ -го входа нейрона

$w_i$  – вес  $i$ -го синапса

Затем определяется значение аксона нейрона по формуле  $Y = f(S)$

Где  $f$  – некоторая функция, которая называется активационной. Наиболее часто в качестве активационной функции используется так называемый сигмоид, который имеет следующий вид:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (7.2)$$

Основное достоинство этой функции в том, что она дифференцируема на всей оси абсцисс и имеет очень простую производную:

$$f'(x) = \alpha f(x)(1-f(x)) \quad (7.3)$$

При уменьшении параметра  $\alpha$  сигмоид становится более пологим, вырождаясь в горизонтальную линию на уровне 0,5 при  $\alpha=0$ . При увеличении  $\alpha$  сигмоид все больше приближается к функции единичного скачка.



Рис. 7.2

Нейронные системы управления относятся к классу нелинейных динамических систем. В составе таких систем искусственная нейронная сеть может выполнять различные функции:

- диагностику технологического оборудования;
- управление подвижными объектами и технологическими процессами;
- прогнозирование ситуаций;
- оценку состояния и мониторинг технологических процессов, и многое другое.

Примеры такого рода нейросетевых систем можно найти во множестве зарубежных и отечественных публикаций.

Применение нейросетевой технологии управления позволяет, в значительной мере, снять математические проблемы математического анализа и синтеза проектируемой системы. Это объясняется тем, что достигаемые свойства и качество процессов управления зависит от фундаментальных свойств многослойных нелинейных нейросетей, а не от аналитически рассчитанных оптимальных законов управления, обычно реализуемых в виде компьютерных программ. Многослойные нейронные сети обладают рядом достоинств, оправдывающих их применение в задачах

управления нелинейными динамическими объектами. Перечислим наиболее существенные из них.

1. Универсальные аппроксимационные свойства многослойных нейронных сетей, играющих ключевую роль в формировании нелинейных алгоритмов управления.

2. Адаптивность нейросетевых структур вследствие их обучения в процессе функционирования.

3. Способность нейросетей к параллельной обработке сигналов, что делает естественным их применение для управления многомерными (многоканальными) объектами управления.

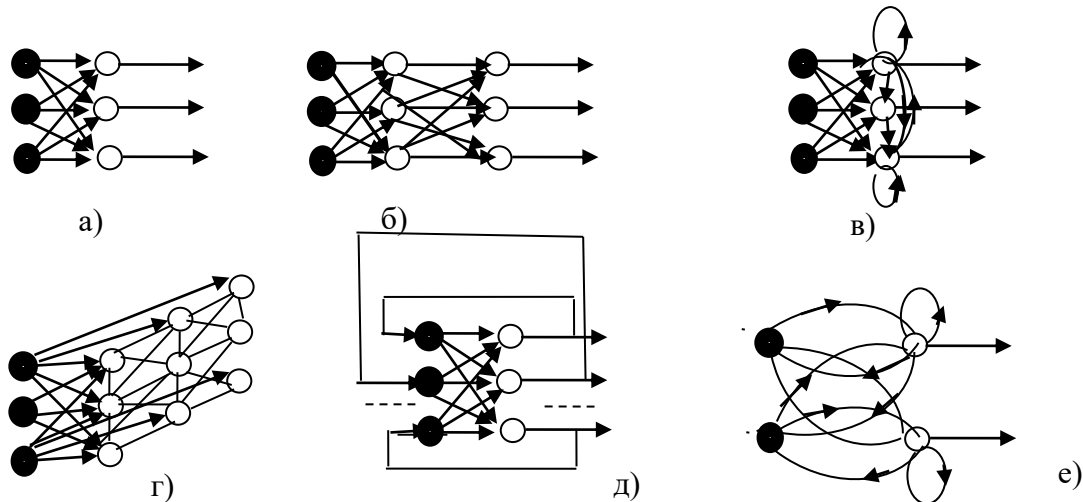


Рис.7.4 Структуры наиболее распространенных ИНС: а) – однослойная сеть; б) – многослойная сеть; в) – сеть с боковыми соединениями; г) – сети

## 7.2. Нейронные сети обратного распространения

Нейронные сети обратного распространения – это мощнейший инструмент поиска закономерностей, прогнозирования, качественного анализа. Такое название – *сети обратного распространения (back propagation)* они получили из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к входному, т. е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

Нейронная сеть обратного распространения состоит из нескольких слоев нейронов, причем каждый нейрон слоя  $i$  связан с каждым нейроном слоя  $i+1$ , т. е. речь идет о полносвязной НС.

В общем случае задача обучения НС сводится к нахождению некой функциональной зависимости  $Y=F(X)$  где  $X$  – входной, а  $Y$  – выходной векторы. В общем случае такая задача, при ограниченном наборе входных данных, имеет бесконечное множество решений. Для ограничения

пространства поиска при обучении ставится задача минимизации целевой функции ошибки НС, которая находится по методу наименьших квадратов:

$$E(w) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (7.4)$$

где  $y_j$  – значение  $j$ -го выхода нейросети;  
 $d_j$  – целевое значение  $j$ -го выхода;  
 $p$  – число нейронов в выходном слое.

Обучение нейросети производится методом градиентного спуска, т. е. на каждой итерации изменение веса производится по формуле:

$$\Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (7.5)$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \cdot \frac{\partial S_j}{\partial w_{ij}}$$

где  $\eta$  – параметр, определяющий скорость обучения;

(7.6)

$y_j$  – значение выхода  $j$ -го нейрона  
 $S_j$  – взвешенная сумма входных сигналов, определяемая по формуле.

При этом множитель

$$\frac{\partial S_j}{\partial w_{ij}} \equiv x_i$$

Где  $x_i$  – значение  $i$ -го входа нейрона.

Далее рассмотрим определение первого множителя формулы (15.6)

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot w_{jk}^{(n+1)}$$

$k$  – число нейронов в слое  $n+1$ .

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \quad (7.7)$$

Тогда мы сможем определить рекурсивную формулу для определения  $\delta_j^{(n)}$   $n$ -ного слоя, если нам известно  $\delta_k^{(n+1)}$  следующего  $(n+1)$ -го слоя.

$$\delta_j^{(n)} = \left[ \sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{dS_j}$$

Нахождение же  $\delta_j^{(n)}$  для последнего слоя НС не представляет трудности, так как нам известен целевой вектор, т. е. вектор тех значений, которые должна выдавать НС при данном наборе входных значений.

$$\delta_j^{(N)} = (y_i^{(N)} - d_i) \cdot \frac{dy_i}{dS_i}$$

И наконец запишем формулу (15.6) в раскрытом виде  $\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot x_i^n$

Рассмотрим теперь полный алгоритм обучения нейросети:

1. Подать на вход НС один из требуемых образов и определить значения выходов нейронов нейросети
2. Рассчитать  $\delta_j^{(N)}$  для выходного слоя НС по формуле (12) и рассчитать изменения весов  $\Delta w_{ij}^{(N)}$  выходного слоя N по формуле (13)
3. Рассчитать по формулам (11) и (13) соответственно  $\delta_j^{(N)}$  и  $\Delta w_{ij}^{(N)}$  для остальных слоев НС,  $n = N-1..1$
4. Скорректировать все веса НС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)$$

1. Если ошибка меньше допустимой, то процесс обучения заканчивается.
2. Если ошибка существенна, то перейти на шаг 1

На этапе 2 сети поочередно в случайном порядке предъявляются вектора из обучающей последовательности.

### 7.3.Повышение эффективности обучения НС обратного распространения

Простейший метод градиентного спуска, рассмотренный выше, очень неэффективен в случае, когда производные по различным весам сильно отличаются. Это соответствует ситуации, когда значение функции S для некоторых нейронов близка по модулю к 1 или когда модуль некоторых

весов много больше 1. В этом случае для плавного уменьшения ошибки надо выбирать очень маленькую скорость обучения, но при этом обучение может занять непоправимо много времени.

Простейшим методом усовершенствования градиентного спуска является введение *момента*  $t$ , когда влияние градиента на изменение весов изменяется со временем. Тогда формула (13) примет вид

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot \delta_j^{(n)} \cdot x_i^n + \mu \Delta w_{ij}^{(n)}(t-1)$$

Дополнительным преимуществом от введения момента является способность алгоритма преодолевать мелкие локальные минимумы.

Основное отличие НС в том, что в них все входные и выходные параметры представлены в виде чисел с плавающей точкой обычно в диапазоне  $[0..1]$ . В то же время данные предметной области часто имеют другое кодирование. Так, это могут быть числа в произвольном диапазоне, даты, символьные строки. Таким образом данные о проблеме могут быть как количественными, так и качественными. Рассмотрим сначала преобразование качественных данных в числовые, а затем рассмотрим способ преобразования входных данных в требуемый диапазон.

#### **7.4. Функциональные структуры нейросетевых систем управления**

Многослойная нейронная сеть прямого действия интерпретируется как *новое технологическое средство формирования функций управления*, соответствующих целевым условиям, моделям объектов и воздействиям внешней среды. Многослойные сети могут быть сопоставлены с оболочкой представления знаний экспертных систем: знаний, например, о динамике нелинейного многосвязного объекта управления или об оптимальном алгоритме управления таким объектом “хранятся” внутри сети. Вопрос состоит в том, *как* извлечь эти знания, используя доступную измерительную информацию в реальном масштабе времени. Это — задача структурного синтеза динамических систем с нейросетями, обучаемыми в темпе поступления данных измерения.

Структурный синтез нейросетевых систем управления базируется на использовании конечного числа типовых динамических структур с многослойными нейронными сетями, дополняемых в каждом конкретном случае необходимыми каналами измерения и оценивания для вычисления переменных,

используемых, в свою очередь, в алгоритмах обучения многослойных нейросетей.

Потенциальная способность сигмоидных нейросетей к обобщенному представлению любого класса непрерывных функций позволяет “конструировать” конкретные типы систем управления, и в первую очередь — для управления многосвязными нелинейными динамическими объектами. Однако применение нейросети взамен типовых регуляторов, рассчитанных на линейные модели объектов, также целесообразно. По сути, использование нейросети инвариантно к классу объектов, если иметь в виду их математическую модель. Определяющим фактором для нейросетевого управления служит доступная для формирования алгоритма обучения/управления текущая информация о состоянии объекта управления и возмущениях внешней среды. Такая информация необходима для формирования *обобщенной ошибки обучения* нейросети для алгоритма ВР.

В данной главе представлены наиболее распространенные функциональные схемы нейросетевых систем идентификации и управления. С применением нейронных сетей в других областях науки и техники можно ознакомиться по публикациям .

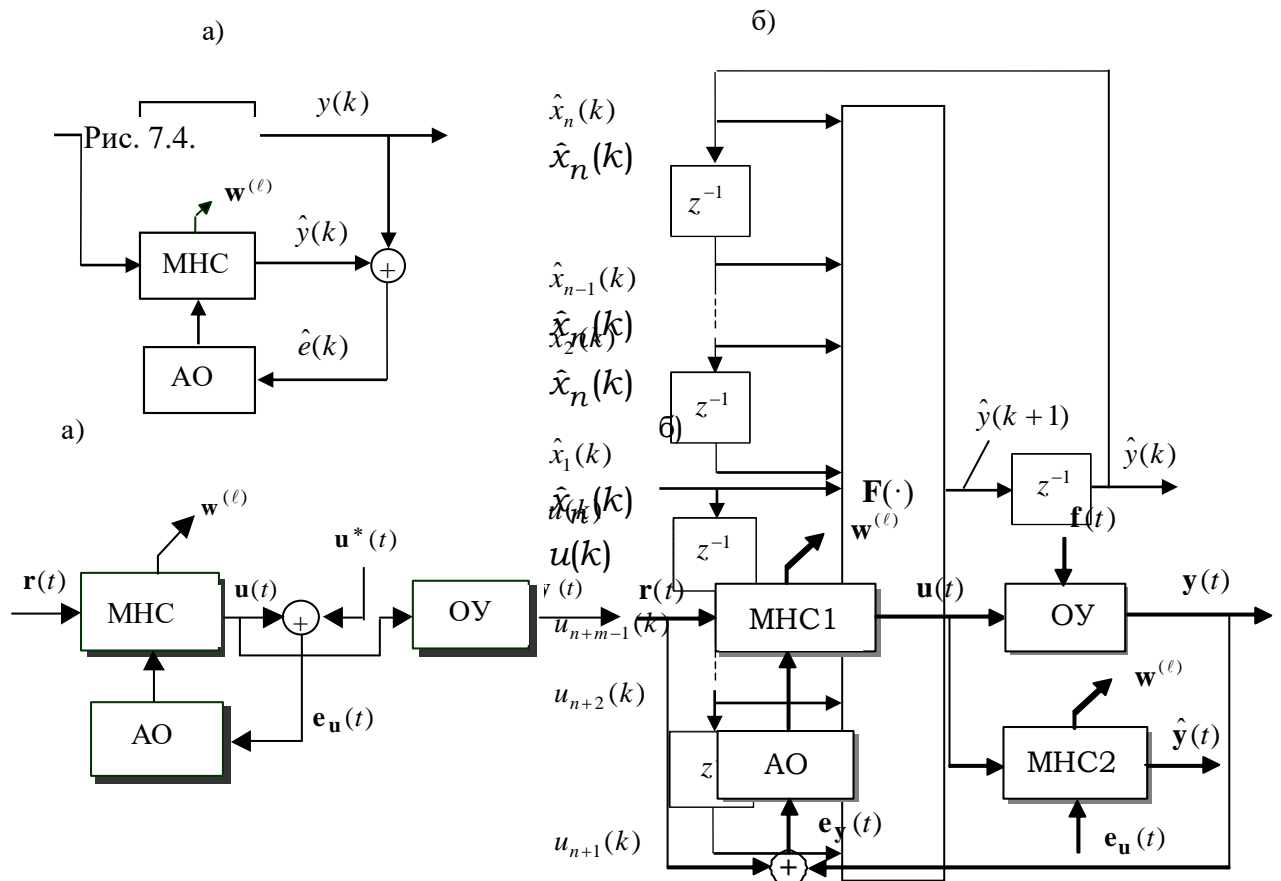
#### 7.5. Типовые модели динамических систем с многослойными нейронными сетями

Рассматриваемые далее типовые модели нейросетевых структур включают элементарные моделирующие и управляющие схемы, которые могут быть базовыми для структурного синтеза функционально более сложных систем управления. Аппроксимирующие способности нейросетей с динамическими алгоритмами обучения позволяют моделировать сложные нелинейные динамические объекты управления — в виде *прямых* и *инверсных* моделей по измерениям “вход—выход” этого объекта. Обе модели используются для вычисления векторов состояния объекта и формирования функции управления объектом.

*Прямая модель обучения.* Известная в теории идентификации схема с настраиваемой моделью представлена на рис. 15.4. Она реализуется в данном случае обучаемой по ошибке  $\hat{e}(t)$  рекуррентной нейросетью:

$$\hat{y}(k+1) = \mathbf{F}(u_k, z^{-1}u_k, \dots, z^{-m}u_k; \hat{y}_k, z^{-1}\hat{y}_k, \dots, z^{-n}\hat{y}_k; \mathbf{w}_i^{(\ell)}).$$

В качестве вектора состояния сети выбран вектор  $\text{col}(y, z^{-1}y, \dots, z^{-n}y) = \text{col}(\hat{x}_n(k), \hat{x}_{n-1}(k), \dots, \hat{x}_1(k))$ , где  $z^{-1}$  — оператор сдвига. Результатом идентификации динамической модели поведения реального ОУ в смысле близости функций выходов  $\hat{y}(t)$  и  $y(t)$  с точностью до ошибки обучения  $\hat{e}(t) = y(t) - \hat{y}(t)$  или минимума функционала  $J(\hat{e}(t))$  могут быть настраиваемые по алгоритму ВР значения весовых коэффициентов  $\mathbf{w}_i^{(\ell)}$  в



слоях  $\ell = \overline{1, K}$  и оценки вектора состояния объекта, описываемого в общем случае параметрически недоопределенным нелинейным дифференциальным уравнением:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1); u(k), \dots, u(k-m+1)].$$

Структурная схема динамической нейросети с одним входом и одним выходом, обучаемой по алгоритму ВР, изображена на рис. 7.4,а. Схема прямой



модели обучения с использованием измерений “вход—выход объекта управления” изображена на рис. 7.4, б. Для получения прямой модели в заданном классе функций  $\mathbf{u}(t) \in \mathbf{U}$  не требуется полной априорной информации о структуре связей и их операторах для ОУ, кроме информации об устойчивости и ограниченности всех траекторий  $\mathbf{y}(t)$  для  $t \geq 0$ . Следует обратить внимание на то, что обученная сеть косвенно “учитывает” влияние на реальный объект управления внешних возмущений  $\mathbf{f}(t)$ .

*Схема обобщенного инверсного обучения.* В изображенной на рис. 7.5,а схеме обобщенного (generalized) инверсного обучения используется тестовый сигнал  $\mathbf{u}^*(t)$ . Это могут быть, например, значения известной функции оптимального управления объектом. Выход реального объекта управления  $\mathbf{y}(t)$  при подключении выхода обученной сети ко входу объекта воспроизводит значения функции  $\mathbf{r}(t)$ , т. е., если  $A$  — оператор ОУ, то при  $\mathbf{e}_{\mathbf{u}}(t) = 0$  нейросеть отображает инверсный оператор  $A^{-1}$ , что и объясняет термин “инверсное обучение”.

Тестовый сигнал  $\mathbf{u}^*(t)$  должен удовлетворять требованию *полного возбуждения*. Обучение многослойной нейросети осуществляется по стандартному алгоритму ВР.

*Схема специализированного инверсного обучения.* Второй подход к инверсному моделированию объекта управления реализуется схемой специализированного инверсного обучения, эта же схема позволяет воспроизводить заданную функцию  $\mathbf{r}(t)$  (см. рис. 7.4, б). В схеме сигнал  $\mathbf{r}(t)$  выполняет роль тестового в задаче инверсного моделирования объекта управления с помощью нейросети (МНС1) и его форма соответствует классу воспроизводимых в задаче управления функций. Схема может содержать вторую нейросеть (МНС2), обученную по схеме рис. 7.2 и применяемую тогда, когда реальный объект находится под воздействием возмущений или не представляется возможным использовать реальный объект для обучения МНС1. В этом случае используется сигнал  $\hat{\mathbf{y}}(t)$ . В статье со ссылкой на работу М. Jordan, D. Rumelhart утверждается, что схема специализированного инверсного обучения позволяет, по крайней мере теоретически, воспроизвести

точную инверсную модель объекта управления при использовании реального выхода  $\mathbf{y}(t)$ . В отличие от обобщенной схемы в специализированной используется ошибка между воспроизводимой функцией  $\mathbf{r}(t)$  и выходом  $\mathbf{y}(t)$ , сеть МНС1 настраивается по динамическому алгоритму обучения. Если нейросеть настроена и ошибка  $\mathbf{e}_{\mathbf{y}}(t) = 0$ , то функция  $\mathbf{r}(t)$  в точности воспроизводится на выходе объекта.

Инверсные модели сами по себе могут быть применимы для управления нелинейными объектами, в том числе, для адаптивного управления. Но прямые и инверсные модели могут служить *типовыми* “строительными блоками” для синтеза многосвязных систем управления нелинейными объектами с более сложной структурой.

Наиболее распространено применение прямой и инверсной обучаемых нейросетевых моделей в нейросетевых системах управления в режиме off-line, когда управления объектом осуществляется в два последовательно выполняемых этапа — этапа идентификации модели объекта и этапа формирования функции управления на входе объекта нейросетевым контроллером с использованием данных идентификации. Наиболее полно такой принцип построения нейросетевых систем управления освещен в обзорной работе.

В основу алгоритмов обучения нейросети и их многочисленных модификаций в этом режиме положен стандартный метод обратного распространения ошибки. Здесь можно выделить две группы алгоритмов обучения:

1) обучение с использованием известной функции управления  $\mathbf{u}^*$ , вычисленной каким-либо аналитическим методом для данного типа нелинейного объекта;

2) обучение по заданной целевой функции с использованием либо имитационной модели объекта, либо текущих измерительных данных непосредственно объекта.

В первой группе алгоритмов используются функции  $\mathbf{u}^*$ , формируемые либо человеком-оператором, либо контроллером-оператором, либо заданные своим множеством значений на конечном интервале времени. Нейросеть повторяет управляющие действия человека-оператора, либо обучающего контроллера, и ее обучение соответствует обучению прямой модели; на входы

сети подаются те же сенсорные данные, что и оператору; целевые выходы сети сравниваются с выработанными человеком (контроллером-оператором) управлениями. Очевидно, что подобное применение сети — так называемый *супервизорный режим* — целесообразно в тех случаях, когда отсутствует аналитическая модель объекта и потому невозможно применение специально рассчитанного автоматического регулятора. Вариант с обучением сети по алгоритму ВР и заданным значениям эталонной функции  $\mathbf{u}^*$  иллюстрируется схемой на рис. 7.4, а.

Вторая группа алгоритмов обучения многослойных нейросетей (как градиентных, так и неградиентных, например, *генетических* алгоритмов) использует для обучения регулирующего контроллера либо идентифицируемую модель объекта, формируемую также нейросетью, либо сам объект с измеряемыми входом и выходом. Применение идентифицирующей нейросети с выходом  $\hat{\mathbf{y}}$  позволяет сформировать ошибку обучения  $\mathbf{e}_y = \mathbf{r} - \hat{\mathbf{y}}$  нейросети, формирующей функцию управления  $\mathbf{q}^{(K)} = \mathbf{u}$ , в тех случаях, когда по каким-либо причинам выход объекта  $\mathbf{y}$  не может быть использован для формирования ошибки обучения.

Прямое инверсное управление использует инверсную модель ОУ, получаемую в соответствии со схемами рис. 7.4, б. Многослойная сеть здесь выполняет функцию регулирующего контроллера. Однако непосредственное применение для обучения сети стандартного алгоритма ВР с аргументом функционала ошибки  $\mathbf{e}_y = \mathbf{r} - \mathbf{y}$ , приводит к неустойчивым процессам в такой системе и необходимо применять динамические алгоритмы обучения типа SBR или BRP. Тем не менее, как это будет показано ниже, применение эталонных моделей системы управления в сочетании с идентифицирующей и регулирующей нейросетями, позволяет применять в двухэтапном режиме управления стандартный алгоритм ВР. Системы с прямым инверсным управлением получили преимущественное применение для управления манипуляторами и “руками” роботов, где задача управления состоит в позиционировании угловых перемещений “рук” для достижения требуемого положения в рабочем пространстве робота.

Рассмотрим ряд схем нейросетевых систем управления, иллюстрирующих приведенную выше классификацию нейросетевых систем управления с обучением нейросетей в режиме off-line.

7.5. Функциональные структуры систем управления  
с многослойными нейросетями

Система управления с прямой и инверсной моделями объекта.

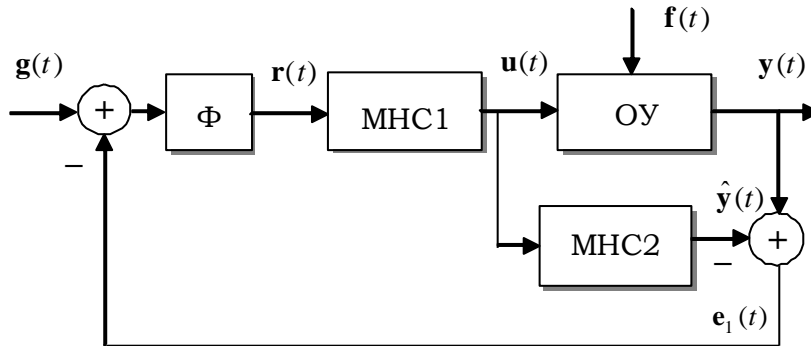
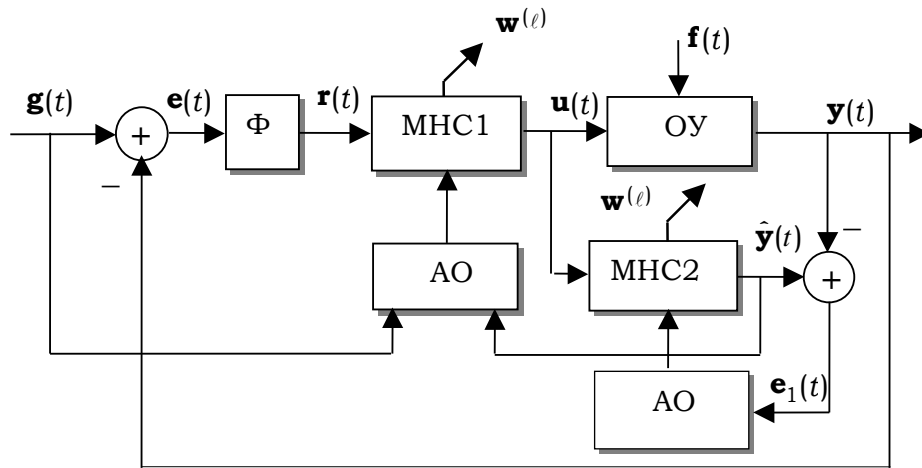


Рис7.5

Изображенная на рис. 7.5 структура в контуре с отрицательной обратной связью исследовалась на устойчивость и грубость для управления нелинейными объектами и известна в теории управления как *система с пассивной адаптацией* или *с косвенным измерением возмущений*. При соответствующем выборе фильтра  $\Phi$  и обученных сетей 1, 2 выходная переменная  $\mathbf{y}(t)$  воспроизводит входной сигнал  $\mathbf{g}(t)$  с желаемой динамикой и не зависит от возмущений  $\mathbf{f}(t)$ , приведенных к выходу объекта управления. Это возможно, если МНС1 отображает инверсный оператор  $A^{-1}$ , а МНС2 — прямой оператор  $A$  объекта управления. Фильтр необходим для достижения же

лаемой динамики системы и его оператор  $A_1 = \tilde{A}^{-1} \{ \hat{A} \}$ , здесь  $\tilde{A}^{-1}$  — реализация МНС1 “точного” инверсного оператора  $A^{-1}$ ,  $\hat{A}$  — реализация МНС2 прямого оператора объекта  $A$ . В этом случае сигнал  $\mathbf{g}(t)$  воспроизводится на выходе объекта без искажений, т. е.  $\mathbf{y}(t) \equiv \mathbf{g}(t)$  и не зависит от возмущения  $\mathbf{v}(t)$ , эквивалентного возмущению  $\mathbf{f}(t)$  и приведенного ко входу объекта. Если прямая и инверсная модели объекта управления отображаются сетями точно, т.е. если  $\tilde{A}^{-1} = A^{-1}$  и  $\hat{A} = A$ , то  $\mathbf{y}(t) = A_1 \{ \mathbf{g}(t) - \mathbf{e}_1(t) \}$ . Вход и выход системы связаны операторным соотношением



$$\mathbf{y}(t) = L^{-1} \{A_1 \{ \tilde{A}^{-1} \{ A \{ \mathbf{g}(t) \} \} \} \},$$

Рис. 7.6.

$$L = 1 + A_1 \{ \tilde{A}^{-1} \{ A \} \} - A_1 \{ \tilde{A}^{-1} \{ \hat{A} \} \}.$$

В этой структуре, в отличие от ранее рассмотренных, возникают проблемы устойчивости и грубости из-за неточной реализации сетями прямой и инверсной моделей объекта управления. Если объект — нелинейный и многосвязный по своей природе и к тому же задан неточной операторной моделью  $A$ , аналитическое решение проблемы едва ли возможно. Поэтому реальный способ исследования свойств системы с прямой и инверсной моделями объекта управления, заданного оператором  $A$ , заключается в компьютерном моделировании. Применение структуры, приведенной на рис. 7.6, ограничено классом стабилизированных объектов из-за отсутствия обратных связей по воспроизводимой функции  $\mathbf{g}(t)$ . Этого недостатка лишена рассматриваемая далее система.

*Адаптивная система управления с прямой и инверсной моделями объекта управления.* Представленная на рис. 7.6. структура адаптивной системы включает две динамические обучаемые нейросети, одна из которых (МНС1) выполняет функции управляющего контроллера. В алгоритме обучения (АО) МНС1 используются текущие измерения воспроизводимой функции  $\mathbf{g}(t)$  и выхода прямой модели объекта  $\hat{\mathbf{y}}(t)$ . Прямая модель выполняется также в виде динамической МНС2, обучаемой по ошибке (прогнозу ошибки)  $\hat{\mathbf{e}}_1(t)$ .

Прямые динамические свойства рассматриваемой системы вытекают из этих рассуждений. В результате обучения МНС2 ошибка  $\mathbf{y}(t) - \hat{\mathbf{y}}(t) \rightarrow 0$  при  $t \geq 0$ , так что  $\hat{\mathbf{y}}(t) \rightarrow \mathbf{y}(t)$ . При настройке МНС1 уменьшается ошибка обучения  $\mathbf{e}(t) = \mathbf{g}(t) - \hat{\mathbf{y}}(t) \rightarrow 0$  при  $t \rightarrow \infty$ . Следовательно,

$\mathbf{y}(t) \rightarrow \mathbf{g}(t)$  и  $\hat{\mathbf{y}}(t) \rightarrow \mathbf{g}(t)$ . Если  $A$  — оператор объекта управления,  $A_c$  — оператор адаптивного контроллера МНС1 и  $A_m$  — оператор модели, реализуемой МНС2, то цепочка предельных переходов при обучении сетей:  $\hat{\mathbf{y}}(t) \rightarrow \mathbf{y}(t) \rightarrow \mathbf{g}(t)$  при  $t \rightarrow \infty$  осуществляется тогда, когда  $A_c = A_m^{-1} = A^{-1}$ . В результате настройки МНС1 отображает инверсную модель объекта управления. Адаптивность системы, изображенной на рис. 6.4, обусловлена минимумом априорной информации об объекте, необходимой для структурного синтеза, и подстройкой модели и контроллера (соответственно МНС2 и МНС1) при неконтролируемых изменениях динамики объекта управления.

*Адаптивная система непрямого действия с прямой и инверсной моделями обучения.* Схема адаптивной системы воспроизведения заданной функции  $r(t)$  на выходе одномерного нелинейного объекта, описываемого разностным уравнением “вход—выход”

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1)] + u(k),$$

изображена на рис.7.6. Уравнение нелинейного объекта управления задано с точностью до параметров в ограниченной области своих значений, поэтому для настройки МНС1 — адаптивного контроллера — используется вектор состояния нелинейного объекта. Для вычисления оценки  $\hat{\mathbf{x}}(k) = \text{col}(\hat{x}_n(k), \hat{x}_{n-1}(k), \dots, \hat{x}_1(k))$  этого вектора используется МНС2, которая, как и МНС1, обучается по методу ВР и ошибке:

$$\begin{aligned} \hat{e}(k) &= y(k) - \hat{y}(k) = f[y(k-1), y(k-2), \dots, y(k-n)] - \\ &- \hat{f}[\hat{x}(k-1), \dots, \hat{x}(k-n)], \quad \hat{x}(k-i) = \hat{x}_{n-i+1}(k), \quad i = \overline{1, n}. \end{aligned}$$

МНС2 реализует прямую модель идентификации объекта управления. Пример такой схемы дан на рис. 7.6. Блоки  $\mathbf{z}^{-n}$  и  $\mathbf{z}^{-m}$  в схеме рис. 7.6 состоят из однотипных звеньев задержки  $z^{-1} = e^{-\Delta t s}$  на период дискретизации  $\Delta t$ . Значения  $n$  и  $m$  соответствуют порядкам левой и правой частей разностного уравнения для объекта управления. Очевидно, что для принятого примера в уравнении объекта вектор коэффициентов  $\beta$  вырожден в скаляр:  $\beta_0 = 1$  и  $m = 0$ . МНС2 выполняет нелинейное преобразование  $f(\cdot) \in \mathbf{F}(\cdot)$  оценок вектора состояния прямой модели объекта управления, настраиваясь по ошибке  $\hat{e}(k)$ . Векторы оценок

$\hat{\mathbf{x}}(k) = \text{col}(\hat{x}_n(k), \hat{x}_{n-1}(k), \dots, \hat{x}_1(k))$  и  $\hat{\mathbf{u}}(k) = \text{col}(\hat{u}_{n+1}(k), \hat{u}_{n+2}(k), \dots, \hat{u}_{n+m-1}(k))$  используются также в адаптивном

контроллере системы — МНС1. Эта сеть также обучается по алгоритму ВР и ошибке обучения  $e_c(k) = y(k) - y_M(k)$ . Эталонная модель (ЭМ) задается линейным разностным уравнением с коэффициентами  $a_i^M$  и  $b_j^M$ , такими, что все нули и полюсы по модулю не превышают единицы:

Рис.15.6 
$$y_M(k+1) = \sum_{i=0}^{n-1} a_i^M x_M(k-i) + \sum_{j=0}^{m-1} b_j^M r(k-j).$$

Нетрудно показать, что алгоритм управления, формируемый обучаемой МНС1 соответствует рекуррентному соотношению

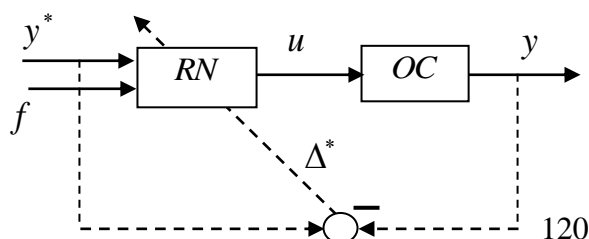
$$u(k) = -\hat{f}[\hat{x}(k), \hat{x}(k-1), \dots, \hat{x}(k-n+1)] + \sum_{i=0}^{n-1} a_i^M x_M(k-i) + \sum_{j=0}^{m-1} b_j^M r(k-j).$$

При этом законе управления и  $k \rightarrow \infty$  достигаются нулевые ошибки обучения МНС2:  $e_c(k) = y(k) - y_M(k)$  и ошибки воспроизведения функции  $r(t)$  и одновременно обучения сети  $\hat{e}(k)$ . Следовательно, МНС1 — инверсная модель нелинейного объекта управления. Обучение нейросетей в структуре, приведенной на рис. 7.4, по текущим измерениям и оценкам вектора состояния объекта делает систему адаптивной по входу  $r(t)$  при неконтролируемых изменениях динамических характеристик объекта и, в частности, параметров его уравнения.

Приведенные примеры нейросетевых систем управления, функционирующих в сетях. Наиболее полно с этим направлением использования нейросетей можно ознакомиться в обзорной статье. Двухэтапный режим обучения/управления позволяет использовать стандартный метод обратного распространения ошибок и различные модификации градиентных вычислительных алгоритмов обратно.

## 7.7 Типовые системы управления, использующих нейросетевые структуры

7.7.1. Последовательная схема нейронной сети и объекта управления (рис. 7.7).



Обучаемая нейросеть соединена последовательно с объектом управления. Информация об управляющем сигнале, а также о

Рис.7.7

контролируемых возмущениях поступает на вход нейронной сети. Нейронная сеть воспроизводит инверсную динамику объекта управления, благодаря чему Обеспечивается равенство управляющего сигнала  $y^*$  и сигнала на выходе объекта управления, так как схема имеет обратную связь по выходу.

7.7.2. Обучение нейронной сети может быть осуществлено различными способами. В частности, обучение в течении работы системы может быть осуществлено таким образом, что сигнал ошибки становится минимальным. Другой способ обучения предполагает введение дополнительной нейронной сети, как показывает рис. 7.8.

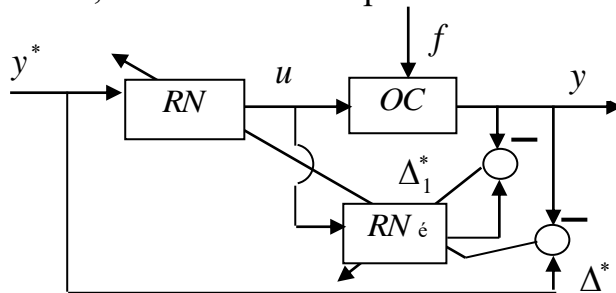


Рис. 15.8.

Эта дополнительная сеть называется эмулятором. Она осуществляет моделирование объекта управления и одновременно обучает управляющую нейронную сеть  $RN$ .

7.7.3. Нейронная сеть включена параллельно обычному устройству управления (рис.7.9). В этом случае НС ( $RN$ ) выполняет роль корректора сигнала управления обычного устройства  $DC$ . Таким образом, сигнал на входе объекта управления является суммой  $u_{DC}$  и  $u_{RN}$ .

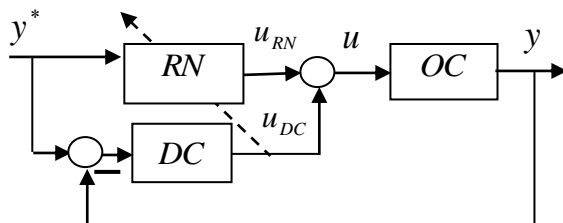


Рис.7.9.

7.4. Схема системы, содержащая нейронную сеть в замкнутом контуре в качестве основного устройства управления (рис. 7.10.)



Нейронная сеть выполняет здесь роль основного устройства управления и обеспечивает хорошее качество управления.

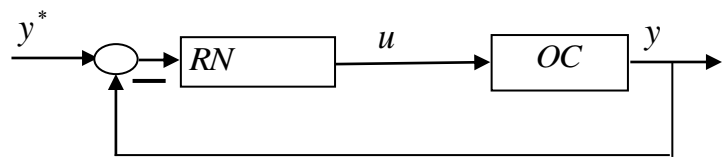


Рис.7.10

Недостатки нейросетевых структур управления :

1) остановка процесса обучения в условиях наличия локального минимума на гиперповерхности эффективности, что приводит к необходимости применять алгоритмы глобальной оптимизации и, как следствие, увеличение времени обучения;

2) невозможность введения в нейронную сеть априорной информации, так как нейронные сети воспринимают новую информацию только в процессе обучения;

3) отсутствие полной теоретической базы, позволяющей выбрать тип структуры нейронной сети, что приводит к необходимости использования алгоритмов самоорганизации и, в конечном итоге, к потерям времени.