

Ю.Н.Хижняков

**Алгоритмы нечеткого, нейронного и нейро-
нечеткого управления в системах
реального времени**

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

«Пермский национальный исследовательский

политехнический университет»

Ю.Н. Хижняков

**Алгоритмы нечеткого, нейронного и
нечетконейронного управления в системах
реального времени**

Утверждено

Редакционно-издательским советом университета

в качестве учебного пособия

Издательство

Пермского национального исследовательского

политехнического университета

2013

УДК 621.311.23:629.12

Хижняков Ю.Н. Алгоритмы нечеткого, нейронного и нейро-нечеткого управления в системах реального времени: учеб. пособие. Пермь: Изд-во ПНИПУ, 2013. – 160 с.

Рассмотрены базовые положения теории нечеткого, нейронного и нечетко-нейронного управления недетерминированными объектами. Разработаны алгоритмы и приведены примеры их применения в системах реального времени.

Предназначено для студентов магистерской подготовки, а также аспирантов специализирующихся в области информационных технологий в проектировании управляющих систем реального времени сетевой образовательной программы магистратуры.

УДК 621.311.23:629.12

© ПНИПУ, 2013

Оглавление

n/n	Наименование тем	
	Введение	6
1. Основы теории нечеткого управления недетерминированными объектами в системах реального времени		10
1.1	Нечеткие множества. Символы. Основные понятия.....	16
1.2	Основные операции над нечеткими множествами: включение, равенство, дополнение, пересечение, объединение, разность, дизъюнктивная сумма. Примеры.....	19
1.3	Лингвистическая переменная. Функции принадлежности. Классификация. Нечеткие числа.	24
1.4.	Фаззификация. Способы реализации.....	31
1.5	Расширение четкой логики <i>И</i> , <i>ИЛИ</i> , <i>НЕ</i>	41
1.6.	Нечеткие отношения. Операции с нечеткими отношениями. Декартово произведение.....	44
1.7	Нечеткая импликация. Варианты реализации.....	50
1.8	Нечеткая композиция. Аналитический способ свертки.....	51
1.9.	Модуль нечеткого логического вывода. Варианты реализации.....	54
1.9.1.	Графическая интерпретация систем нечеткого логического вывода	56
1.10.	Дефаззификация. Способы реализации.....	71
1.11.	Алгоритм нечеткого управления в системах реального времени.....	75
	Контрольные вопросы.....	84

2. Основы теории нечетко-нейронного управления недетерминированными объектами в системах реального времени			86
2.1.	Биологический нейрон. Основные положения теории искусственных нейронных сетей.....		86
2.2.	Функции активации.....		92
2.3	Теоремы Колмогорова-Арнольда и Хехт-Нильсона.		95
2.4.	Разновидности и классификация нейронных сетей.....		96
2.4.1.	Нейронные сети без обратной связи		98
2.5.	Алгоритмы обучения нейронной сети с учителем.....		98
2.5.1.	Алгоритм обратного распространения ошибки.....		98
2.5.2.	Генетические алгоритмы.....		105
2.6.	Адаптивный нейрон. Алгоритмы адаптации нейрона.....		113
2.7.	Нечеткая нейронная продукционная сеть <i>Anfis</i>		127
2.7.1.	Нечеткая нейронная продукционная сеть <i>Anfis</i> с применением алгоритма Сугено-Такаги.....		127
2.7.2.	Нечеткая нейронная продукционная сеть <i>Anfis</i> с применением алгоритма Такаги-Сугено-Канга.....		139
2.7.3.	Нечеткая нейронная продукционная сеть <i>Anfis</i> с применением алгоритма Ванга-Менделя		145
2.8.	Модификация нечеткой продукционной сети <i>Anfis</i> и ее разновидностей с применением адаптивных нейронов.....		149
	Контрольные вопросы.....		
	Заключение		
	Литература.....		

Введение

В настоящее время высшая школа Российской Федерации выполняет переход на уровневую систему высшего профессионального образования, регламентируемую принятым Законом «Об образовании в РФ» [1]. Введенные в 2011 г. новые Федеральные государственные образовательные стандарты высшего профессионального образования (ФГОС ВПО) устанавливают новые требования и вносят существенные изменения в реализуемые основные образовательные программы (ООП), обусловленные, в первую очередь, введением компетентного подхода в организацию образовательного процесса, основанного на деятельностной модели подготовки выпускника вуза.

В связи с этим проведение лекционных занятий по дисциплине основывается на активном методе обучения, при котором учащиеся являются активными участниками занятия, отвечающие на заранее намеченный преподавателем список вопросов, стимулирующих ассоциативное мышление и установления связей с ранее освоенным материалом. Практические занятия проводятся на основе реализации метода обучения действием: определяются проблемные области; формируются группы для их решения; каждое практическое занятие проводится по своему алгоритму. Проведение семинарских занятий основывается на интерактивном методе обучения, при котором учащиеся взаимодействуют не только с преподавателем, но и друг с другом. При этом доминирует активность учащихся в процессе обучения. Место преподавателя в интерактивных семинарских занятиях – направление деятельности магистрантов на достижение целей занятия.

Сетевой характер дисциплины «Алгоритмы нечеткого, нейронного и нейро-нечеткого управления в системах реального времени» допускает веб-занятия: дистанционные лекции, семинары, практические занятия. Они используют специализированные образовательные форумы, на которых пользователи взаимодействуют по определённой теме или проблеме с помощью записей, оставляемых на одном из сайтов с установленной на нем

соответствующей программой. Это дает возможность многодневной работы с асинхронным характером взаимодействия магистров и преподавателя.

Учебный курс «Алгоритмы нечеткого, нейронного и нейро-нечеткого управления в системах реального времени» рассматривает построение алгоритмов с максимальным быстродействием и возможность их реализации. Применение машин фон Неймана с последовательной процедурой расчетов требует переосмысливания, т.е. наступает эра распараллельных, многопроцессорных вычислительных устройств, связанных с развитием электроники [2]. К таким вычислительным средствам можно отнести нейрокомпьютеры, у которых основной операционный блок построен на основе нейронной сети и реализуется нейросетевыми алгоритмами.

Основная идея создания нейронной ЭВМ является как аналого-цифровой, где «быстрая» – аналоговая часть выполняла многомерные операции в пороговом базисе, а «медленная» – в цифровом виде. Создание нейронных ЭВМ требует построения новых алгоритмов решения многомерных задач. Основным аппаратом построения нейронных алгоритмов является теория нейронных сетей, где базисом для большинства задач является базис $\{\sum a\bar{x}\}$, к таким задачам относятся: задачи векторной алгебры, преобразования Фурье, задачи оптимизации, решение дифференциальных уравнений, уравнений Пуассона, Эйлера, эллиптических уравнений и пр.

В настоящее время в виду отсутствия нейрокомпьютеров необходимое распараллеливание с целью повышения производительности вычислений можно обеспечить применением систем реального времени (СРВ). В СРВ правильность функционирования зависит не только от логической корректности, но и от времени, за которое эти вычисления производятся. Реальное время есть программное обеспечение (продукт), утвержденное международным стандартом IEEE 610.12 – 1990 [3].

Реальное время относится к системе или режиму работы, в котором вычисления производятся в течение времени, определяемого внешним процессором. Основное требование к СРВ заключается в том, чтобы эти системы должны выполнять свои операции вовремя. В системах реального времени быстроедействие управления адекватно скорости протекания физических явлений на регулируемом объекте. Увеличение времени, необходимого на опрос датчиков и времени формирования управляющего воздействия, может оказать отрицательное влияние на качество управления в случае задач реального времени.

СРВ предсказуемо (в смысле времени реакции) реагируют на не предсказуемые (по времени появления) внешние события. Предсказуемость или детерминированность СРВ позволяет разработчику обеспечить функциональность и корректность спроектированной системы. При этом скорость реакции системы важна только относительно скорости протекания внешних процессов, за которыми СРВ должна следить, или которыми должна управлять. СРВ призваны решать задачи за определенное выделенное время, максимальное его значение называется критическим сроком обслуживания.

Задачи реального времени по виду ограничений делятся:

– на задачи «жесткого» реального времени, когда запросы на формирование управляющего воздействия в контуре должны строго выполняться. Понятие задачи «жесткого» реального времени исключает в принципе невыполнение задачи за критический срок обслуживания. Нарушение критических сроков обслуживания может привести к значительному материальному ущербу, к трагедии, например, в авиации. Для проверки, что такая система соответствует своим временным ограничениям, применяют расширенное моделирование и т.п.

– задачи мягкого реального времени, когда запросы не всегда выполняются, но при планировании вероятность этих нарушений было бы уменьшена. Задачи «мягкого» реального времени допускают нарушение

выполнение задачи или за выход критического срока обслуживания. Это нарушение критических сроков обслуживания может привести к незначительному материальному ущербу, например, требует дополнительной обработки сигнала;

- задачи с постоянной периодичностью появления запросов и их длительностью;

- задачи с динамическими изменяемыми параметрами (период измерения, обработка измеряемых параметров и т.п.).

В СРВ необходимо знание резерва количества свободного времени, для оценки которого целесообразно использовать аппарат нечеткой логики для систем автоматического управления объектами с неопределенным и определенным математическим описанием. К таким системам можно отнести системы с недетерминированным управлением, т.е. системы с нечетким, нейронным и нечетко-нейронным управлением. Применения систем реального времени к системам с неполным описанием объекта регулирования возможно, так как процедура обработка информации внутри таких систем абсолютно нейтральна и все требования, которые выдвигают системы реального времени здесь необходимо строго выполнять. Рассмотрим более подробно реализацию нечеткого, нейронного и нейро-нечеткого управления.

1. Основы теории нечеткого управления недетерминированными объектами в системах реального времени

Теория нечетких множеств и нечеткой логики представляет собой обобщение и переосмысление классической математики. У ее истоков лежат идеи:

- многозначной логики, которая указала на возможности перехода от двух к произвольному числу значений истинности и обозначила проблему оперирования понятиями с изменяющимся содержанием;
- теории вероятностей, которая, породив большое количество различных способов статистической обработки экспериментальных данных, открыла пути определения и интерпретации функций принадлежности;
- дискретной математики, которая предложила инструмент для построения моделей многомерных и многоуровневых систем, удобных для решения практических задач.

Существует достаточное количество промышленных объектов, характеризующихся существенной нелинейностью, и, зачастую, отсутствием приемлемой аналитически заданной модели. Если некоторые процессы объекта управления плохо поддаются формализации и математическому описанию, то в существующей системе управления используют регулятор нечёткой логики параллельно традиционному регулятору, т.е. нечеткая логика используется для замены традиционных алгоритмов управления и совместно с ними. При использовании методов каскадного управления с классическим регулятором во внутреннем контуре регулятор нечёткой логики может быть применен во внешнем контуре.

Экспериментально показано, что нечеткое управление дает лучшие результаты, по сравнению с результатами, получаемыми при классических

алгоритмах управления. Очевидной областью внедрения алгоритмов нечеткой логики являются всевозможные экспертные системы, в том числе:

- нелинейный контроль и управление в производстве (домна, робот, сушилка и т.д.);
- самообучающиеся системы;
- системы, распознающие тексты на естественном языке;
- системы планирования и прогнозирования, опирающиеся на неполную информацию;
- финансовый анализ в условиях неопределенности;
- управление базами данных.

Нечеткие числа, получаемые в результате «не вполне точных измерений», во многом аналогичны распределениям теории вероятностей, но по сравнению с вероятностными методами, методы нечеткой логики позволяют резко сократить объем производимых вычислений, что, в свою очередь, приводит к увеличению быстродействия нечетких систем.

Исходя из вышесказанного, на рис. 1-1 показаны области наиболее эффективного применения современных технологий управления.

Как видно, классические методы управления хорошо работают при полностью управляемом и наблюдаемом объекте управления и детерминированной среде, а для систем с неполной или неточной информацией и высокой сложностью объекта управления оптимальными являются нечеткие методы управления.

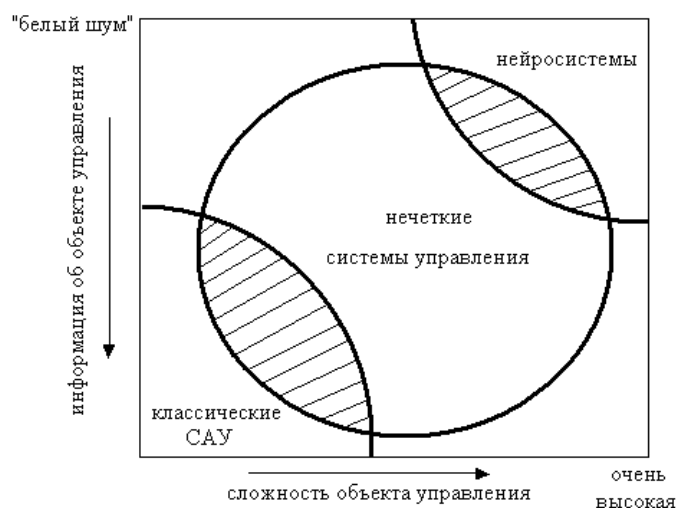


Рис.1-1. Области наиболее эффективного применения современных систем управления

В правом верхнем углу рис.1-1 приведена еще одна современная технология управления – с применением искусственных нейронных сетей, которая позволяет обеспечить высокие показатели качества при высокой сложности объекта и практически полном отсутствии данных о нём.

Подводя итоги анализа применения нечеткой логики в системах управления, можно увидеть, что наилучшие результаты для систем средней сложности и детерминированности информационной среды достигаются применением нечетких алгоритмов управления, а для систем высокой сложности и малой детерминированности применением нейросетей. Однако не исключено совместное применение, как классических методов управления, так и методов искусственных нейронных сетей с нечеткими системами.

При разработке нечетких систем необходимо пройти следующие этапы проектирования (после изучения основных понятий нечетких множеств и систем):

- определить входы и выходы создаваемой системы;
- задать для каждой из входных и выходных лингвистических переменных терм-множества функции принадлежности;
- разработать базы правил выводов для реализуемой нечёткой системы;
- провести дефаззификацию;

– провести настройку и анализ адекватности разработанной модели в системе реального времени.

Как отличить сущность нечеткой логики от теории вероятности?

Пример 1-1. Какую выбрать бутылку для утоления жажды?

Вариант *A*: задана вероятность событий $P = 0,9$, т.е. 9 из 10 есть чистая вода, а 1 из 10 – кислота, т.е. пить нельзя и табличка может находиться в стороне от бутылей.

Вариант *B*: задана функция принадлежности $\mu = 0,9$, т.е. 0 – кислота, 1 – чистая вода, а 0,9 – болотная вода, т.е. ее можно пить после кипячения и табличка должна находиться у бутылки с болотной водой.

Анализ вариантов показывает, что в варианте *B* функция принадлежности показывает лишь степень принадлежности элемента $x \in A$ и не является вероятностью, поскольку $\sum \mu_A(x) \neq 1$. Степени принадлежности по существу отражают порядок расположения элементов в нечетком подмножестве *A*.

Нечеткие системы тесно связаны с нейронными сетями и генетическими алгоритмами, образуя новые вычислительные технологии. В частности, генетические алгоритмы можно применять для подбора синапсов, а также для формирования базы правил и функций принадлежности нечеткой системы. В свою очередь нейронные сети позволяют выбирать соответствующие параметры для самих генетических алгоритмов (параметры скрещивания и мутации) и определения скорости обучения нейронной сети, т.е. саму философию нейронных сетей можно заложить в основание нечетких систем, которые обретают способность к обучению и адаптации (гибридные системы).

Символы теории четких и нечетких множеств

\forall – для всех

\in – принадлежит;

\leftrightarrow – эквивалентность;

- \exists – существует;
- \subset – строгое включение;
- \subseteq – включение
- \setminus – разность множеств;
- \cap – пересечение;
- \cup – объединение;
- \triangleq – равно по определению;
- \equiv – тождественно равно;
- $=$ – равенство;
- $\{ \}$ – множество;
- $*$ – алгебраическое произведение;
- \check{A} – носитель A ;
- \circ – композиция отношений;
- $+$ – дизъюнктивная сумма;
- \rightarrow – импликация отношений (логическая связка);
- \neg – дополнение;
- $[]$ – отрезок.

Естественный язык формировался в течение сотен лет как средство общения и как структура, отражающая объективный мир. Познание мира опирается на мышление, а мышление, в свою очередь, невозможно без определенной знаковой системы (естественный язык). Естественный язык способен оперировать противоречивыми, сложными и многозначными понятиями. В ходе принятия решения человек легко овладевает ситуацией, разделяя ее на события, находит решения в сложных ситуациях с помощью правил принятия решения. Язык, который использует человек – это нечеткий естественный язык. Полученная модель системы не является унифицированной: она либо описывает свойства фрагментов объекта, либо является набором нескольких локальных моделей, поставленных в определенные условия. Сами локальные модели не используют числовые значения. Обладая некоторой общностью, они просты для понимания на

естественном уровне. Актуальность новой технологии – нечеткое моделирование, обусловленное тенденцией увеличения сложности математических моделей в системе реального времени. В настоящее время наблюдается интенсивное развитие и практическое применение нечетких систем для целей управления и регулирования многих технических объектов. Нечёткая логика появилась в 1965 г. в работах Лотфи А. Заде, проф. технических наук Калифорнийского университета в г. Беркли. Концепция нечеткого множества зародилась у Заде «как неудовлетворенность математическими методами классической теории систем, которая вынуждала добиваться искусственной точности, неуместной во многих системах реального мира, особенно в так называемых гуманистических системах, включающих людей». С ростом сложности системы постепенно падает способность человека делать точные и в то же время значащие утверждения относительно ее поведения, пока не будет достигнут порог, за которым точность и значимость становятся взаимоисключающими характеристиками. Впервые теория нечетких множеств и нечеткой логики была применена в Англии. В 1973 г. Эбрахим Мамдани из Лондонского колледжа Королевы Мэри применил для управления паровым двигателем нечеткий контроллер. В Японии первый нечеткий контроллер разработал Сугено (*Sugeno*) в 1983 г. для очистки воды. На Востоке нечеткая логика пошла на «ура», так как для людей, воспитанных на восточной философии, с ее неоднозначными и расплывчатыми категориями, нечеткая логика сразу стала своей, родной. В 1994 г. Коско (*Kosko*) доказал теорему о нечеткой аппроксимации, согласно которой, любая математическая система может быть аппроксимирована системой на нечеткой логике. Следовательно, с помощью естественно-языковых высказываний «ЕСЛИ – ТО», с последующей их формализацией средствами теории нечетких множеств, можно отразить произвольную взаимосвязь «выход–вход» без использования сложного аппарата дифференциального и интегрального исчислений, традиционно применяемого в управлении и идентификации. Сегодня нечеткая логика

рассматривается как стандартный метод моделирования и проектирования. В 1997 г. язык нечеткого управления внесен в Международный стандарт программируемых контроллеров IEC 1131-7. Несмотря на то, что математический аппарат нечёткой логики впервые был разработан в США, активное развитие данного метода началось в Японии, и новая волна вновь достигла США и Европы. Изобретены следующие устройства: фотоаппарат, видеокамеры, стиральные машины, автомобильные трансмиссии, кондиционеры, тостеры, пылесосы, приборы экономии электроэнергии и водоснабжения, интеллектуальные датчики. Применяются в компьютерах для регулирования частоты разгона, в управлении движением в метро в г. Сиднее и т.д.

Аппарат нечеткой логики (*fuzzy logic*) применяется для решения задач, в которых исходные данные являются ненадежными и слабо формализуемыми. Математическая теория нечетких множеств позволяет описывать нечеткие понятия и знания, оперировать этими описаниями и делать нечеткие выводы. Нечеткое управление полезно, когда исследуемые процессы являются сложными для анализа с помощью общепринятых методов. Нечеткая логика, представляющая средство отображения неопределенностей и неточностей реального мира, ближе к человеческому мышлению и естественным языкам, чем традиционное логическое мышление. Основной причиной появления новой теории стало наличие нечётких и приближённых рассуждений при описании человеком процессов, систем, объектов. Нечёткая логика является многозначной логикой, что позволило определить промежуточные значения для оценок да / нет, истинно / ложно и т. д. Выражения подобные таким, как «слегка тепло» или «довольно холодно» возможно формулировать математически и обрабатывать на компьютерах. Нечеткие методы, основанные на теории нечетких множеств характеризуются: использованием лингвистических переменных вместо числовых переменных или дополнение к ним; простые

отношения между переменными описываются с помощью нечетких высказываний; сложные отношения описываются нечеткими алгоритмами.

1.1. Нечеткие множества. Символы. Основные понятия

Подход к формализации понятия «нечеткого множества» состоит в обобщении понятия принадлежности [4]. В обычной теории множеств существует несколько способов задания множества. Одним из них является задание универсального множества E с помощью характеристической функции, где E состоит, например из подмножества целых чисел, подмножества гладких функций и т.д. Четкое подмножество $A \in E$ имеет четкие границы, элементы которого удовлетворяют свойству G и определяется как подмножество упорядоченных пар

$$A = [\mu_A(x)/x],$$

где $\mu_A(x)$ – характеристическая функция, принимающая значение 1, если x удовлетворяет свойству G , и 0 – в противном случае.

$$\mu_A(x) = \begin{cases} 1, & \text{если } x \in A; \\ 0, & \text{если } x \notin A. \end{cases}$$

Особенностью этой функции является бинарный характер ее значений.

Нечеткое подмножество не имеет четкой границы и отличается тем, что для элементов $x \in E$ нет однозначного ответа «да-нет» относительно свойства G . В связи с этим нечеткое подмножество $A \in E$ определяется как подмножество упорядоченных пар

$$A = \{\mu_A(x)/x\},$$

где $\mu_A(x)$ – функция принадлежности, принимающая вещественные значения в некотором вполне упорядоченного множества положительных вещественных чисел (единичный интервал) $M = [0, 1]$. Могут быть и другие интервалы, например $[1, 100]$.

В случае непрерывного множества E можно использовать следующее обозначение

$$A = \int_E \mu_A(x) / x,$$

где знак \int обозначает совокупность пар $\mu_A(x)/x$.

Пример 1-2. Пусть $E = \{x_1, x_2, x_3, x_4, x_5\}$; $M = [0, 1]$; A – нечеткое подмножество, для которого $\mu_A(x_1) = 0,3$; $\mu_A(x_2) = 0$; $\mu_A(x_3) = 1$; $\mu_A(x_4) = 0,5$; $\mu_A(x_5) = 0,9$. Тогда подмножество A можно представить: $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$ или $A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5$, где знак $+$ имеет смысл объединения.

Нечеткие множества есть естественное обобщение обычных множеств, когда мы отказываемся от бинарного характера характеристической функции и предполагаем функцию принадлежности, которая может принимать любые значения на отрезке $[0, 1]$. В теории нечетких множеств характеристическая функция называется функцией принадлежности, а ее значение $\mu(x)$ – степенью принадлежности элемента x нечеткому подмножеству A . Более строго, нечетким подмножеством A называется совокупность пар, которые можно записать в разных вариантах:

$$A = \{x, \mu_A(x)\} = \{\mu_A(x), x\} = \left\{ \frac{\mu_A(x)}{x} \right\} = \left\{ \frac{x}{\mu_A(x)} \right\} = \{x/\mu_A(x)\} = \{\mu_A(x)/x\} = \int_E \mu_A(x)/x.$$

Основные характеристики нечеткого множества

1. Величина $\sup \mu_A(x)$ называется высотой A . Если $\sup \mu_A(x) = 1 \quad \forall x \in E$, то подмножество A нормально, если $\sup \mu_A(x) < 1 \quad (x \in E)$, то подмножество A – субнормально.

2. Нечеткое множество A пусто, если $\forall x \in E \quad \mu_A(x) = 0$. Непустое субнормальное множество можно нормализовать по формуле

$$\mu_A(x) = \frac{\mu_A(x)}{\sup \mu_A(x)} \quad x \in E.$$

(Читается: функция принадлежности элемента x подмножеству A).

3. Нечеткое множество унимодально: $\mu_A(x) = 1$ только на одном x из E .

4. Элементы $x \in E$, для которых $\mu_A(x) = 0,5$ называются точками перехода подмножества A .

5. Носитель нечеткого множества A (основание) называется нечеткое множество A таких точек в E , для которых $\mu_A(x) > 0$ или $\text{supp } A = \{x/\mu_A(x) > 0\}, \forall x \in E$.

Пример 1-3. Пусть $E = \{0, 1, 2, \dots, 10\}, M = [0, 1]$.

Нечеткое множество под названием «несколько» можно определить следующим образом: «несколько» = $0,5/3 + 0,8/4 + 1/5 + 1/6 + 0,8/7 + 0,5/8$.

Его характеристики: высота $A = 1$; носитель = $\{3, 4, 5, 6, 7, 8\}$; точки перехода = $\{3, 8\}$.

Пример 1-4. Пусть $E = \{-8, -5, -3, 0, 1, 2, 4, 6, 9\}, M = [0, 1]$.

Нечеткое подмножество чисел по абсолютной величине «близкой к нулю» можно определить, например, так:

$$A = \{0/-8 + 0,5/-5 + 0,6/-3 + 1/0 + 0,9/1 + 0,8/2 + 0,6/4 + 0,3/6 + 0/9\}.$$

Его характеристики: $A = 1$; носитель = $\{0, 1, 4, 6, 9\}$; точка перехода = $\{-5\}$.

В теории управления универсальное множество E обычно связывают с некоторыми физическими величинами: температура $t^{\circ}\text{C}$, давление P , расход Q и т.д., которые получают с измерительных датчиков установленных на регулируемом объекте, которые описываются естественным языком. Учитывая разные характеристики лингвистических переменных (например, температура воды может быть теплой, горячей), вводится понятие термина, а всю совокупность терм называют терм-множеством. Терм можно выразить графически либо аналитически через функции принадлежности, которые подчеркивают, что термы есть подмножества, показанное на рис. 1-2. Будем ассоциировать с подмножеством A диапазон изменения температуры от 0°C до 50°C , а с подмножеством B её изменение в диапазоне 50°C до 100°C . В лингвистической интерпретации терму A будет соответствовать «холодная вода», а терму B – «горячая вода». Таким образом, весь температурный диапазон можно представить в виде терм-множества из двух терм.

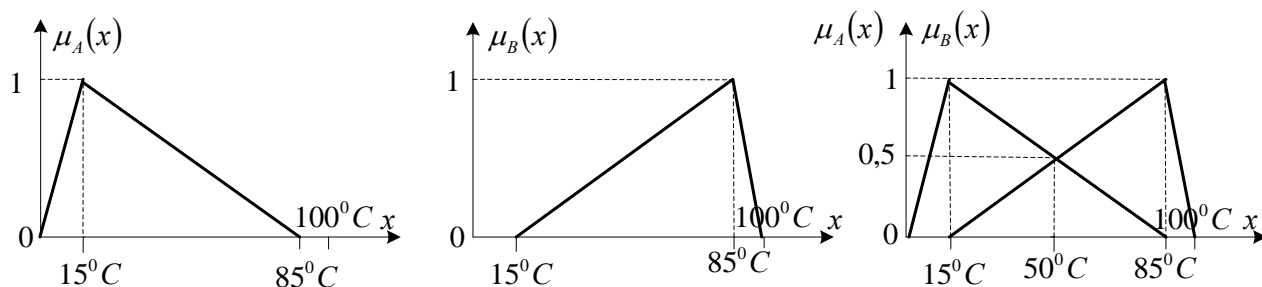


Рис.1-2. К построению терм-множества

1.2. Основные операции над нечеткими множествами

1. Включение (доминирование, операция вложение).

Пусть A и B – нечеткие подмножества на универсальном множестве E .

Говорят, что A содержится в B , если $\forall x \in E \quad \mu_A(x) \leq \mu_B(x)$.

Обозначение: $A \subset B$. Иногда используют термин "доминирование", когда $A \subset B$, говорят, что B доминирует A .

2. Равенство (эквивалентность).

Нечеткие подмножества A и B равны, если $\forall x \in E \quad \mu_A(x) = \mu_B(x)$.

Обозначение: $A = B$.

3. Дополнение.

Пусть $M = [0,1]$, A и B – нечеткие подмножества заданы на E .

Если A и B дополняют друг друга, то обозначается $A = \bar{B}$ или $\mu_A(x) = 1 - \mu_B(x)$ и обозначается $B = \bar{A}$ или $\mu_B(x) = 1 - \mu_A(x)$.

Очевидно, что $\bar{\bar{A}} = A$. (Дополнение определено для $M = [0,1]$, но очевидно, что его можно определить для любого упорядоченного M).

4. Пересечение.

$A \cap B$ – наибольшее нечеткое подмножество, содержащееся одновременно в A и B с функцией принадлежности: $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ или операция над множествами A и B , результатом которой является множество C , которое содержит те элементы, которые принадлежат и множеству A и множеству B .

5. Объединение.

$A \cup B$ – наименьшее нечеткое подмножество, включающее как A , так и B , с функцией принадлежности: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ ИЛИ операция над подмножествами A и B , результатом которой является подмножество C , которое содержит те элементы, которые принадлежат подмножеству A , или подмножеству B , или обоим подмножествам.

6. Разность.

$A - B = A \cap \bar{B}$ с функцией принадлежности:

$$\mu_{A-B}(x) = \mu_{A \cap \bar{B}}(x) = \min(\mu_A(x), 1 - \mu_B(x)).$$

7. Дизъюнктивная сумма.

$A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$ с функцией принадлежности:

$$\mu_{A \oplus B}(x) = \max\{\min\{\mu_A(x), 1 - \mu_B(x)\}; \min\{1 - \mu_A(x), \mu_B(x)\}\}$$

Пример 1-5. Пусть

$$A = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4;$$

$$B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4;$$

$$C = 0,1/x_1 + 1/x_2 + 0,2/x_3 + 0,9/x_4.$$

Здесь:

1. $A \subset B$, т.е. A содержится в B или B доминирует A , C несравнимо ни с A , ни с B , т.е. пары $\{A, C\}$ и $\{B, C\}$ – пары недоминируемых нечетких множеств.

2. $A \neq B \neq C$.

3. $\bar{A} = 0,6/x_1 + 0,8/x_2 + 1/x_3 + 0/x_4$; $\bar{B} = 0,3/x_1 + 0,1/x_2 + 0,9/x_3 + 0/x_4$.

4. $A \cap B = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4$.

5. $A \cup B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4$.

6. $A - B = A \cap \bar{B} = 0,3/x_1 + 0,1/x_2 + 0/x_3 + 0/x_4$;

7. $B - A = \bar{A} \cap B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4$.

8. $A \oplus B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4$.

На основе операций алгебраического произведения определена операция возведения в степень положительного числа α нечеткого

подмножества A . Нечеткое подмножество A^α определяется функцией принадлежности $\mu_{A^\alpha} = \mu_A^\alpha(x)$. Частным случаем возведения в степень являются:

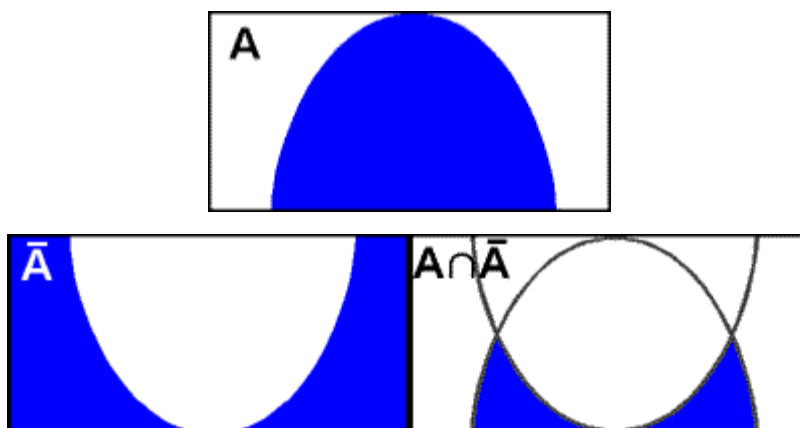
- $CON(A) = A^2$ – операция концентрирования (уплотнения) (более чем);
- $ДИЦ(A) = A^{0,5}$ – операция растяжения (почти что).

Аналогично часто используют степени 4 и 0,25, интерпретируя их как $A^4 = \text{более чем более } A$, т. е. $A = (\text{более чем})^2 A$ и т.д. Операции уплотнения и растяжения необходимы для построения ручного регулятора коррекции терм и др.

Графическое представление нечетких множеств

Для нечетких подмножеств можно строить визуальное представление. Рассмотрим прямоугольную систему координат, на оси ординат которой откладываются значения $\mu_A(x)$, на оси абсцисс расположены в произвольном порядке элементы E (мы уже использовали такое представление в примерах нечетких множеств). Если E по своей природе упорядочено, то этот порядок желательно сохранить в расположении элементов на оси абсцисс. Такое представление делает наглядными простые операции над нечеткими подмножествами, показанными на рис. 1-3. На верхней части рисунка заштрихованная часть соответствует нечеткому подмножеству A и, если говорить точно, изображает область значений A и всех нечетких подмножеств, содержащихся в A .

На нижней части рисунка – даны \bar{A} , $A \cap \bar{A}$, $A \cup \bar{A}$.



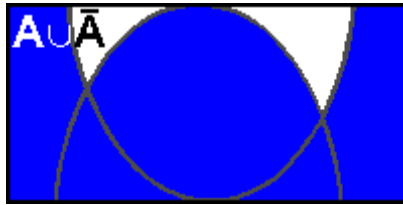


Рис.1-3. Наглядное представление операций над нечеткими подмножествами

Пример 1-6. Пусть A – нечеткое подмножество «от 5 до 8» и B – нечеткое подмножество «около 4» заданы на рис.1-4. Необходимо показать пересечение, объединение и дополнение множеств.

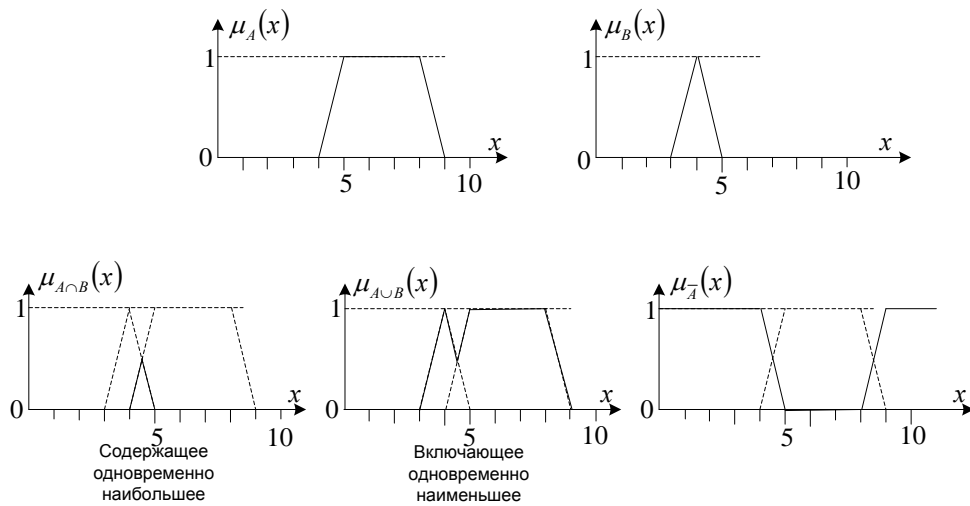


Рис.1-4. Наглядное представление операций над нечеткими множествами

Пример 1-7. Пусть $X = R$, где R – подмножество действительных чисел, то подмножество действительных чисел «близких числу 7», можно определить функцией принадлежности вида

$$\mu_A(x) = \frac{1}{1 + (x - 7)^2},$$

а нечеткое подмножество действительных чисел «близких к 7» описывается выражением

$$A = \int_x \left[\frac{1}{1 + (x - 7)^2} \right] dx.$$

Пример 1-8. Пусть температура воды в море задана подмножеством $X = \{15^\circ, \dots, 25^\circ\}$. Выразить с помощью нечетких подмножеств оценку температуры воды в море.

Первый отдыхающий, лучше всего чувствующий себя при температуре 21°C , определил для себя нечеткое подмножество:

$$A = \frac{0,1}{15} + \frac{0,3}{17} + \frac{0,5}{18} + \frac{0,8}{19} + \frac{0,95}{20} + \frac{1}{21} + \frac{0,9}{22} + \frac{0,8}{23} + \frac{0,75}{24} + \frac{0,7}{25}.$$

Второй отдыхающий, предпочитающий температуру 20°C , определил для себя следующее:

$$B = \frac{0,1}{15} + \frac{0,2}{16} + \frac{0,7}{18} + \frac{0,9}{19} + \frac{1}{20} + \frac{0,9}{21} + \frac{0,85}{22} + \frac{0,8}{23} + \frac{0,75}{24} + \frac{0,7}{25}.$$

С помощью нечетких подмножеств A и B показано неточное определение температуры купания в море.

Свойства операций пересечения \cap и объединения \cup .

Пусть A , B и C – нечеткие подмножества. Им свойственно следующее:

- 1) $\begin{cases} A \cap B = B \cap A \\ A \cup B = B \cup A \end{cases}$ – коммутативность;
- 2) $\begin{cases} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cup B) \cup C = A \cup (B \cup C) \end{cases}$ – ассоциативность;
- 3) $\begin{cases} A \cap A = A \\ A \cup A = A \end{cases}$ – идемпотентность;
- 4) $\begin{cases} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{cases}$ – дистрибутивность;
- 5) $A \cup \emptyset = A$, где \emptyset – пустое множество, т.е. $\mu_{\emptyset}(x) = 0 \forall x \in E$

$$A \cap \emptyset = \emptyset$$

- 6) $A \cap E = A$, где E – универсальное множество;

$$A \cup E = E$$

- 7) $\begin{cases} \overline{A \cap B} = \bar{B} \cup \bar{A} \\ \overline{A \cup B} = \bar{B} \cap \bar{A} \end{cases}$ – теоремы де Моргана

Замечание. Введенные выше операции (пересечение и объединение) над нечеткими подмножествами основаны на использовании операций *max* и *min*. В теории нечетких подмножеств разрабатываются вопросы построения обобщенных, параметризованных операторов пересечения, объединения и дополнения, позволяющих учесть разнообразные смысловые оттенки соответствующих им связок "и", "или", "не".

1.3. Лингвистические переменные. Функции принадлежности.

Классификация. Нечеткие числа

Лингвистическая переменная принимает значения из множества слов или словосочетаний некоторого естественного языка. Понятие лингвистической переменной играет важную роль в нечетком логическом выводе и в принятии решений на основе приближенных рассуждений. Формально лингвистическая переменная описывается следующей пятеркой:

$$\langle x, T, E, G, M \rangle,$$

где x – имя переменной; T – терм-множество, каждый элемент которого задается нечетким подмножеством на универсальном множестве E ; G – синтаксические правила, порождающие названия значений переменной или задающие функции принадлежности нечетких термов; M – семантические правила, задающие физический смысл функции принадлежности нечетких термов.

Пример 1-9. Рассмотреть с помощью лингвистической переменной u «температура в комнате».

Температуру в комнате можно выразить через универсальное множество E с терм-множеством $T = \{\langle \text{«холодно»}, \langle \text{«комфортно»}, \langle \text{«жарко»} \rangle\}$ с такими, например, функциями принадлежности:

$$\mu_{\text{холодно}}(x) = \frac{1}{1 + \left| \frac{x - 12^0}{6} \right|^{12}}; \mu_{\text{юморно}}(x) = \frac{1}{1 + \left| \frac{x - 20^0}{3} \right|^8}; \mu_{\text{жарко}}(x) = \frac{1}{1 + \left| \frac{x - 35^0}{8} \right|^{12}}; x \in E;$$

Синтаксические правила G порождают новые термы с использованием квантификаторов: не, очень, более-менее и т.д.

Семантические правила M , заданы в табл.1-1.

Функции принадлежности

Существуют разные варианты высказываний о функции принадлежности в литературе:

1. Функция принадлежности нечеткого подмножества A приписывает каждому элементу $x \in X$ степень принадлежности к нечеткому подмножеству A [4].

2. Функция принадлежности указывает степень принадлежности элементов x подмножеству A .

Таблица 1-1. Семантические правила

Квантификатор	Функция принадлежности
t	$\mu_t(x)$
Не t (не тепло)	$\mu = 1 - \mu_t(x)$
Очень t (очень тепло)	$\mu = (\mu_t(x))^2$
Более-менее t (более-менее тепло)	$\mu = \sqrt{\mu_t(x)}$

3. Функция принадлежности определяет характер термы. Численное значение функции принадлежности – степень принадлежности элемента $x \in X$.

4. Функция принадлежности используется для представления лингвистических переменных в виде нечетких подмножеств.

Функции принадлежности можно строить по результатам экспертных данных.

Пример 1-10. Построить функции принадлежности термов «низкий», «средний», «высокий», используемых для лингвистической оценки переменной «рост мужчины» [6]. Результаты опроса пяти экспертов сведены в табл.1-2.

Таблица 1-2. Рост мужчины

	Терм	[160, 165)	(165, 170)	(170, 175)	(175, 180)	(180, 185)	(185, 190)	(190, 195)	(195, 200]+
Эксперт 1	Низкий	1	1	1	0	0	0	0	0
	Средний	0	0	1	1	1	0	0	0
Эксперт 2	Низкий	1	1	1	0	0	0	0	0
	Средний	0	0	1	1	0	0	0	0
Эксперт 3	Низкий	1	0	0	0	0	0	0	0
	Средний	0	1	1	1	1	1	0	0
Эксперт 4	Низкий	1	1	1	0	0	0	0	0
	Средний	0	0	0	1	1	1	0	0
Эксперт 5	Низкий	1	1	0	0	0	0	0	0
	Средний	0	1	1	1	0	0	0	0

Результаты обработки экспертных мнений сведены в табл.1-3. Графики функций принадлежности показаны на рис.1-5.

Функции принадлежности полностью описывают нечеткие подмножества и определяют степень принадлежности элемента x нечеткому подмножеству, называемому в форме числовых значений в диапазоне $[0, 1]$. Функции принадлежности лингвистических переменных, как правило, перекрывают друг друга, поэтому для одной и той же лингвистической переменной эти функции могут сообщать различные степени принадлежности лингвистических термов, отличающихся от нуля.

Таблица 1-3. Обработка результатов экспертных мнений

Терм	160-165	165-170	170-175	175-180	180-185	185-190	190-195	195-200
Низкий	5	4	3	0	0	0	0	0

	1	0,8	0,6	0	0	0	0	0
Средний	0	2	4	5	3	2	0	0
	0	0,4	0,8	1	0,6	0,4	0	0
Высокий	0	0	0	1	2	4	5	5
	0	0	0	0,2	0,4	0,8	1	1

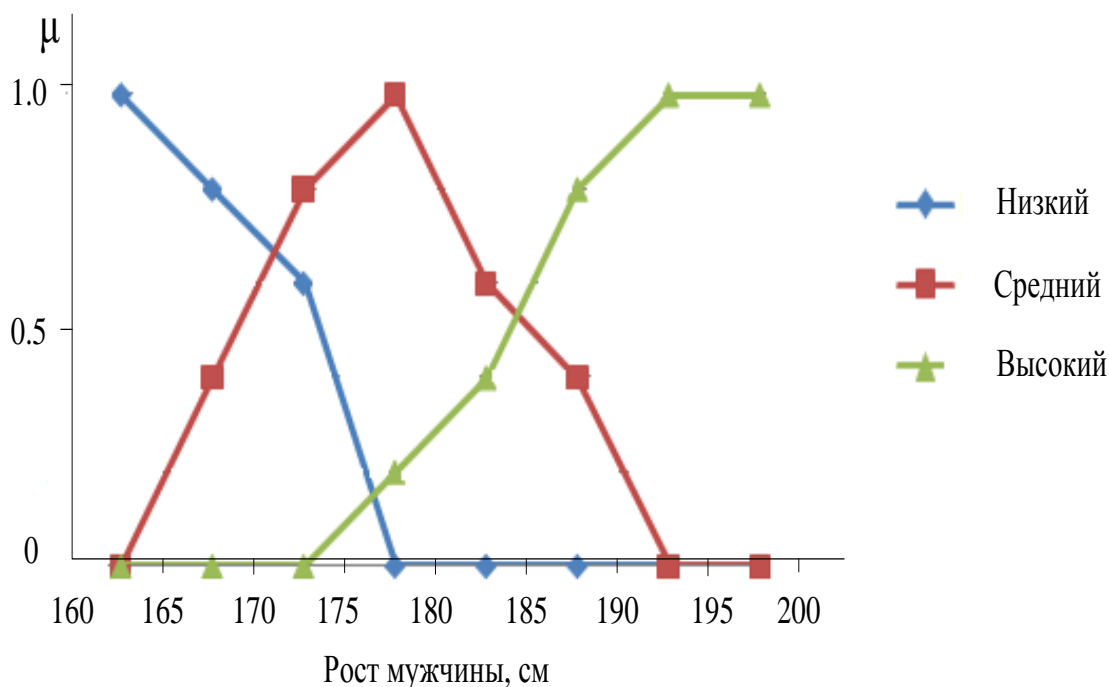


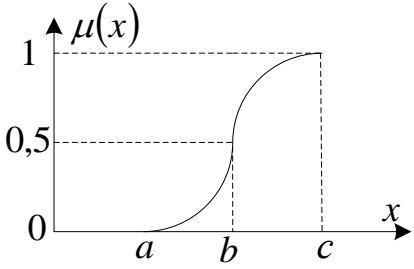
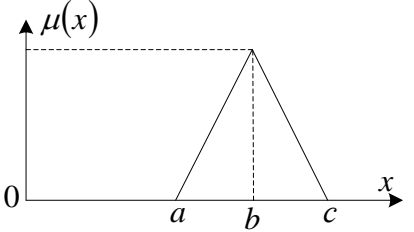
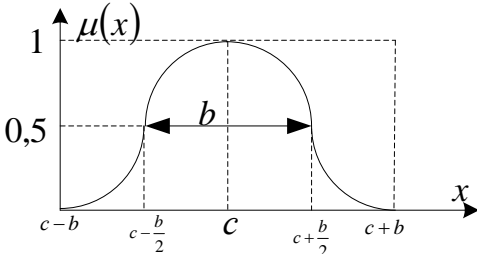
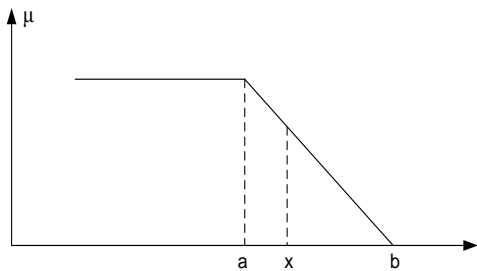
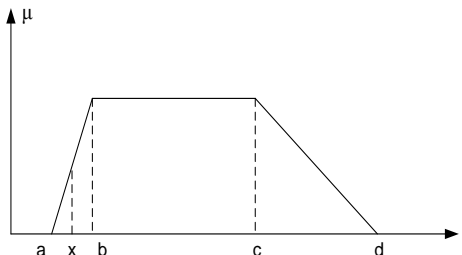
Рис. 1-5. Функции принадлежности нечетких множеств

Функции принадлежности бывают линейные и нелинейные, которые приведены в табл. 1-4.

Таблица 1-4

Функции принадлежности

ФП	График	Аналитическое выражение
Класса γ		$trn(x, a, b) = \begin{cases} 0, & \text{if } (x \leq a); \\ \frac{x-a}{b-a}, & \text{if } (a < x \leq b); \\ 1, & \text{if } (x > b); \end{cases}$

Класса S		$trn(x, a, v, c) = \begin{cases} 0, & \text{if } x \leq a, \\ 2\left(\frac{x-a}{c-a}\right)^2, & \text{if } a \leq x \leq b, \\ 1 - 2\left(\frac{x-c}{c-a}\right)^2, & \text{if } b \leq x \leq c, \\ 1, & \text{if } x \geq c, \end{cases}$ <p>где $v = (a+c)/2$</p>
Класса T		$trn(x, a, b, c) = \begin{cases} 0, & \text{if } (x \leq a); \\ \frac{x-a}{b-a}, & \text{if } (a < x \leq b); \\ \frac{c-x}{c-b}, & \text{if } (b \leq x < c); \\ 0, & \text{if } (x \geq c). \end{cases}$
Класса π		$\pi(x, b, c) = \begin{cases} s\left(x, c-b, c-\frac{b}{2}, c\right) & \text{для } x \leq c \\ 1 - s\left(x, c, c+\frac{b}{2}, c+b\right) & \text{для } x > c \end{cases}$ <p>при $x = c \pm \frac{b}{2} = 0,5$</p>
Класса L		$trn(x, a, b) = \begin{cases} 1, & \text{if } (x \leq a); \\ \frac{b-x}{b-a}, & \text{if } (a < x \leq b); \\ 0, & \text{if } (x > b); \end{cases}$
Класса П		$trn(x, a, b, c, d) = \begin{cases} 0, & \text{if } (x \leq a); \\ \frac{x-a}{b-a}, & \text{if } (a < x \leq b); \\ 1, & \text{if } (b \leq x < c); \\ \frac{d-x}{d-c}, & \text{if } (c \leq x < d); \\ 0, & \text{if } (x \geq d); \end{cases}$

Синглето н		$\mu_A(x) = \begin{cases} 1, & x_i = x; \\ 0, & x_i \neq x \end{cases}$ $\mu_A(x) = \sin gl(x - x_i)$
---------------	--	---

PS. Синглетон имеет следующие определения: одноэлементное множество; одиночная функция принадлежности; унимодальное нечеткое множество на одном $x \in E$; четкая пара $(x, \mu(x))$, где $\mu(x) = 1$.

Классификация функций принадлежности

На рис.1-6 показана классификация функций принадлежности.

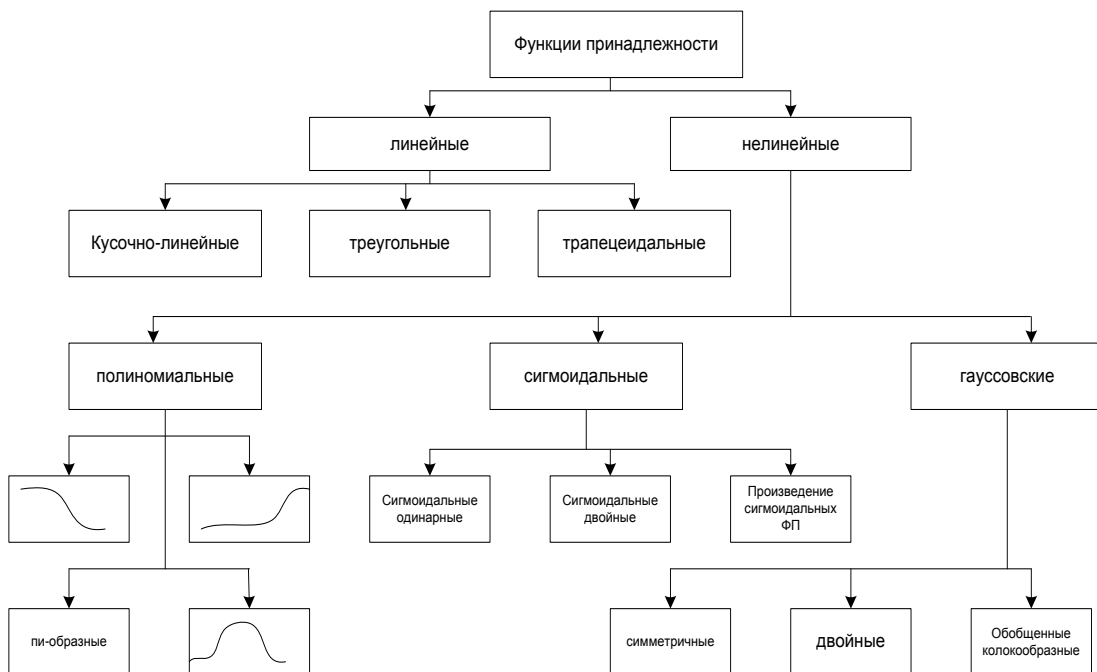


Рис. 1-6. Классификация функций принадлежности

Колоколообразная функция (разновидность функции класса π принадлежности имеет вид

$$f(x) = \exp\left(-\frac{(x-m)^2}{\sigma^2}\right),$$

где x – аргумент функции принадлежности класса π ; m – значение нечеткой переменной, при которой достигается наибольшее значение функции принадлежности (матожидание); σ – среднеквадратическое отклонение (СКО) функции принадлежности от максимального значения (матожидания).

Нечеткие числа

Нечеткие числа – нечеткие переменные, определяемые на числовой оси, т.е. нечеткое число определяется как нечеткое подмножество A на множестве действительных чисел с функцией принадлежности $\mu_A(x) \in [0,1]$, где $x \in A$. Нечеткое число A нормально, если $\max \mu_A(x) = 1$.

Пример 1-11. Пусть универсум E есть множество действительных чисел. Задать нечеткое подмножество A , обозначающее множество чисел, близких к 10 аналитически.

Возможно задать подмножество A функцией принадлежности аналитически:

$$\mu_A(x) = (1 + |x - 10|^n)^{-1}, \text{ где } n \in \mu_A(x),$$

где показатель степени n выбирается в зависимости от степени близости к 10. Например, для описания множества чисел «очень близких к 10» можно положить $n = 4$; для множества чисел, не очень близких от 10, $n = 1$.

1.4. Фаззификация. Способы реализации

Система управления с нечеткой логикой оперирует нечеткими подмножествами. В задачах управления чаще всего применяется операция фаззификации на синглетонной базе. Блок фаззификации содержит от одного до нескольких терм-множеств. Поэтому конкретному значению $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^T \in X$ входного сигнала блока нечеткого управления после операции фаззификация будет сопоставлено нечеткое подмножество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$. Переменным сигналом блока фаззификации является унимодальная функция принадлежности (синглетон), которая синхронно с входным сигналом перемещается в нормированном интервале изменения входной величины, пересекая терм-множество.

Фаззификация на синглетонной базе (операция перевода четкой информации в нечеткую информацию) в случае нечеткого управления формирует нечеткое подмножество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$ для дальнейшей обработки операцией нечеткая импликация в блоке выработки решения. Число терм отдельного терм-множества задается от 3 до 11 в зависимости от скорости изменения сигнала. Увеличение числа терм возможно включением квантификаторов: НЕ, ОЧЕНЬ, БОЛЕЕ-МЕНЕЕ. Характер терм задаются синтаксическими правилами, а смысл – семантическими правилами.

Фаззификатор нечеткого регулятора преобразует четкую входную информацию в нечеткую информацию (формирует нечеткое подмножество).

Формирования нечеткого подмножества на выходе фаззификатора выполняется:

- на синглетонной базе, где терм-множество располагается в нормированном интервале от минус единицы до плюс единицы;
- на несинглетонной базе, где терм-множество располагается в нормированном интервале от минус единицы до плюс единицы и синглетон заменяется треугольной функцией принадлежности;
- на базе пропорционального пересчета входного параметра терм-множества с основанием равным нормированному интервалу $0 \dots +1$ [7];
- на базе замены терм-множества сигмоидных функций принадлежности сетью из искусственных нейронов [5].

Рассмотрим различные формы фаззификации

Фаззификация на синглетонной базе с основанием терм-множества в нормированном интервале $-1 \dots +1$

Для формирования нечеткого подмножества на синглетонной базе используется унимодальная функция принадлежности (синглетон), который жестко связан с входной переменной. Для случая двух лингвистических переменных зададимся следующими терм-множеством

Степени принадлежности проекции вектора x_1 после фаззификации

$$\mu_{A_1^1}(x_1) = \delta(x_1^1 - \bar{x}_1)$$

$$\mu_{A_1^2}(x_1) = \delta(x_1^2 - \bar{x}_1)$$

$$\mu_{A_1^3}(x_1) = \delta(x_1^3 - \bar{x}_1)$$

$$\mu_{A_1^4}(x_1) = \delta(x_1^4 - \bar{x}_1)$$

$$\mu_{A_1^5}(x_1) = \delta(x_1^5 - \bar{x}_1)$$

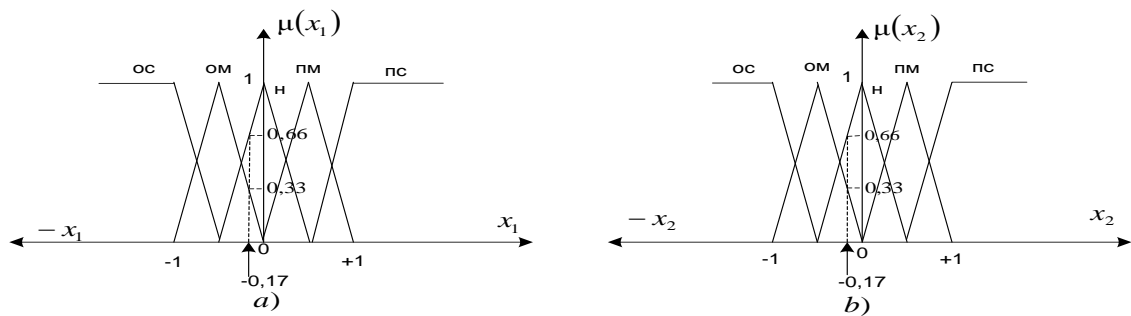


Рис.1-7. Терм-множество фаззификатора: а – терм-множество лингвистической переменной x_1 ; б – терм-множество лингвистической переменной x_1 .

Степени принадлежности проекции вектора x_2 после фаззификации

$$\mu_{A_2^1}(x_2) = \delta(x_2^1 - \bar{x}_2)$$

$$\mu_{A_2^2}(x_2) = \delta(x_2^2 - \bar{x}_2)$$

$$\mu_{A_2^3}(x_2) = \delta(x_2^3 - \bar{x}_2)$$

$$\mu_{A_2^4}(x_2) = \delta(x_2^4 - \bar{x}_2)$$

$$\mu_{A_2^5}(x_2) = \delta(x_2^5 - \bar{x}_2)$$

Фаззификация на несинглетонной базе с основанием терм-множества в нормированном интервале -1...+1

На рис. 1-8 приведена иллюстрация графической интерпретация нечеткого вывода по Мамдани, где фаззификация выполнена на не синглетонной базе. Синглетоны представлены треугольными функциями

принадлежности $A_1^1 A_2^1 A_1^2 A_2^2$, которые синхронно перемещаются с четкими входными сигналами.

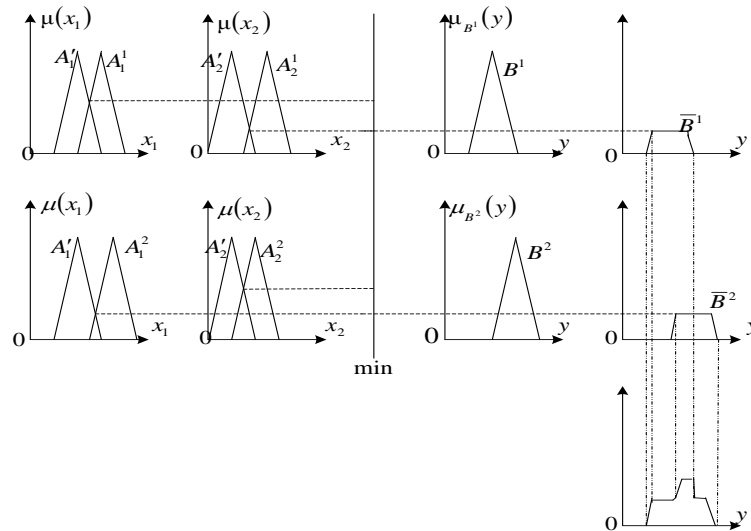


Рис.1-8. Графическая интерпретация нечеткого вывода по Мамдани, где фаззификация реализована на несинглетонной базе.

Фаззификация на базе пропорционального пересчета входного параметра терм-множества с основанием равным нормированному интервалу 0...+1

Для формирования нечеткого подмножества используется пропорция, пересчитывающая фиксированное значение входного параметра в соответствующий элемент $u^* \in [0,1]$ согласно рис.1-9, где очевидна следующая пропорция [7]

$$\frac{(x_{bi} - x_{hi})}{(1-0)} = \frac{(x_i^* - x_{hi})}{(u^* - 0)}.$$

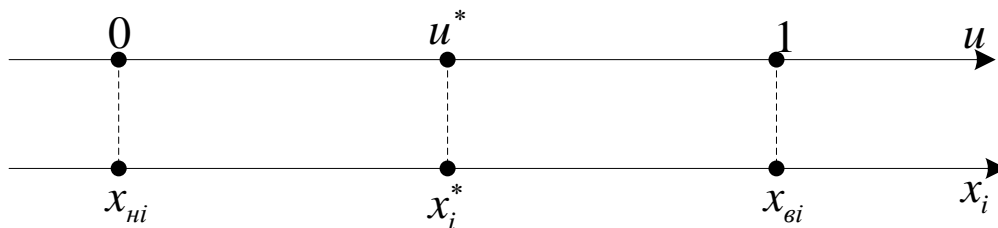


Рис.1-9. К вопросу определения пропорции пересчета входного параметра

Тогда

$$\mu^* = \frac{(x_i^* - x_{hi})}{(x_{vi} - x_{hi})}, \quad (1-1)$$

где x_i^* – текущее значение входного параметра;

x_{vi} – единичный (верхний) уровень нормированного интервала;

x_{hi} – нулевой уровень нормированного интервала;

μ^* – элемент, пропорциональный входному параметру.

Рассмотрим пример формирования функций принадлежности с основанием, равным нормированному интервалу.

На рис.1-10 приведено терм-множество из пяти терм с одним основанием.

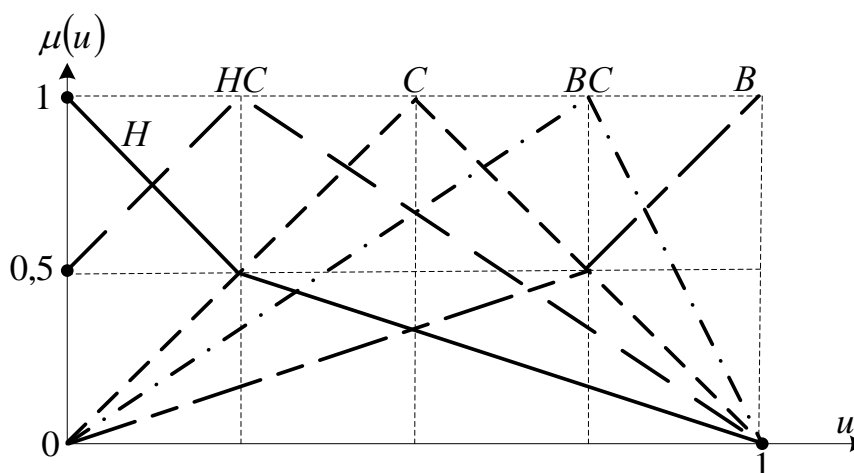


Рис.1-10. Графическое задание терм-множеств с одним основанием: Н – низкий; HC – ниже среднего; C – средний; BC – выше среднего; В – высокий

Запишем аналитические выражения терм-множества (см. рис.1-10), воспользовавшись координатами (u_1, μ_1) и (u_2, μ_2) , которые заложены в уравнении прямой

$$\mu(u) = \frac{[(\mu_2 - \mu_1)u + \mu_1 u_2 - \mu_2 u_1]}{(u_2 - u_1)}.$$

Тогда аналитические выражения терм-множества будут иметь вид:

$$\mu_H(u) = \begin{cases} 1 - 2u, & \text{где } u \in \left[0, \frac{1}{4}\right]; \\ \frac{2(1-u)}{3}, & \text{где } u \in \left[\frac{1}{4}, 1\right]; \end{cases}$$

$$\mu_{HC}(u) = \begin{cases} 0,5 + 2u, & \text{при } u \in \left[0, \frac{1}{4}\right]; \\ 1,5 - 2u, & \text{при } u \in \left[\frac{1}{4}, \frac{1}{2}\right]; \\ 1 - u, & \text{при } u \in \left[\frac{1}{2}, 1\right]; \end{cases}$$

$$\mu_C(u) = \begin{cases} 2u, & \text{при } u \in \left[0, \frac{1}{2}\right]; \\ 2(1-u), & \text{при } u \in \left[\frac{1}{2}, 1\right]; \end{cases}; \quad \mu_{BC}(u) = \begin{cases} u, & \text{при } u \in \left[0, \frac{1}{2}\right]; \\ 2u - \frac{1}{2}, & \text{при } u \in \left[\frac{1}{2}, \frac{3}{4}\right]; \\ \frac{5}{2} - 2u, & \text{при } u \in \left[\frac{3}{4}, 1\right]; \end{cases};$$

$$\mu_B(u) = \begin{cases} \frac{2u}{3}, & \text{при } u \in \left[0, \frac{3}{4}\right]; \\ 2u - 1, & \text{при } u \in \left[\frac{3}{4}, 1\right]; \end{cases}.$$

Переход от полученных функций принадлежности терм-множества $\mu^T(u)$ к функциям принадлежности $\mu^T(x)$ определяется соотношением (1-1). Для настройки $\mu^T(u)$ можно пользоваться операцией возведения в степень: $[\mu(u)]^C$, где показатель степени определяет изменение формы функции принадлежности. Применение операций сжатия и растяжения можно осуществлять для каждого отрезка функции принадлежности, как показано на рис.1-11. Коэффициент C называется коэффициентом относительной важности. При различном числе термов функции принадлежности аппроксимируют треугольными термами, которые строятся с соблюдением правил:

- основанием треугольника является универсум $U_i = \{0, L_i - 1\}$, где L_i – целое число, соответствующее количеству термов лингвистической переменной;

- термы нумеруются целыми числами от 1 до L_i ;

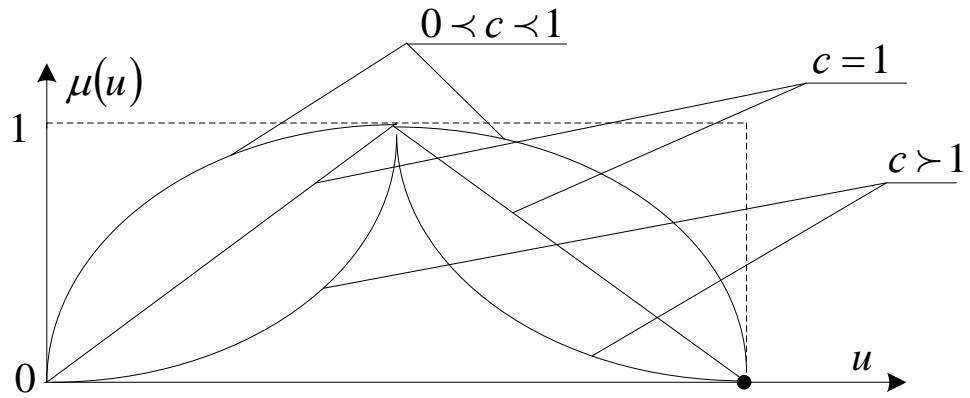


Рис.1-11. Влияние коэффициента относительной важности на линейную функцию принадлежности

– вершина треугольника соответствует номеру терма.

Диапазон $[x_{hi}, x_{bi}]$ изменения входного параметра x_i отображают на универсум $U_i = \{0, L_i - 1\}$. Пересчет фиксированного значения входной переменной $x_i^* \in [x_{hi}, x_{bi}]$ в соответствующий элемент $u_i^* \in [0, L_i - 1]$ определяется пропорцией

$$\frac{(x_{bi} - x_{hi})}{(L_i - 1)} = \frac{(x_i^* - x_{hi})}{u_i^*},$$

из которой получим

$$u_i^* = \frac{(L_i - 1)(x_i^* - x_{hi})}{(x_{bi} - x_{hi})}.$$

Функция принадлежности терма с номером j , показанная на рис. 1-12, определяется прямыми линиями, которые проходят через точки с координатами:

$$[0,1] \text{ и } [L_i - 1,0] \text{ при } u \in [0, L_i - 1];$$

$$[0,0] \text{ и } [j-1,1] \text{ при } u \in [0, j-1] \text{ и } [j-1,1] \text{ и } [L_i - 1,0] \text{ при } u \in [j-1, L_i - 1] \text{ для } j = 2, L_i - 1;$$

$$[0,0] \text{ и } [L_i - 1,1] \text{ при } u \in [0, L_i - 1] \text{ для } j = L_i.$$

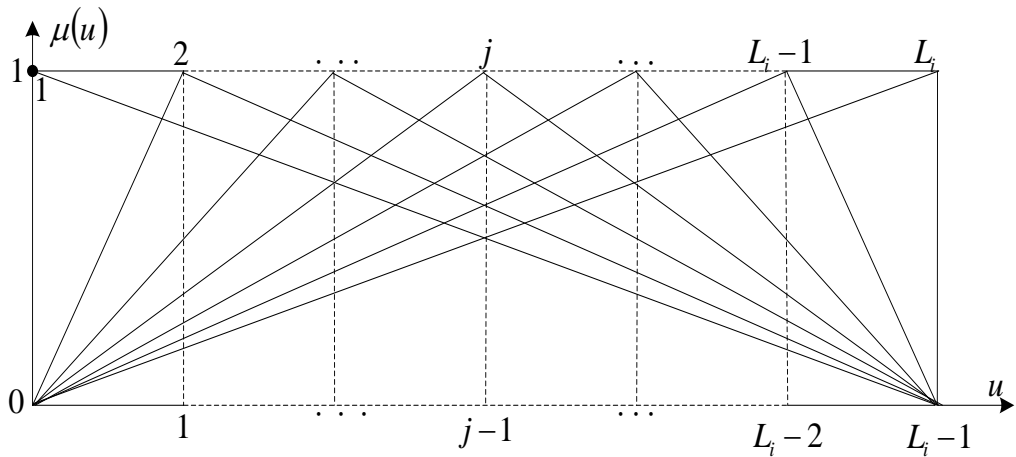


Рис.1-12. К построению j -ых терм с одним основанием

Используя выражение для u^* , проходящей через две точки с известными координатами, получим

$$\mu_j(u) = \begin{cases} 1 - \frac{u}{(L_i - 1)}, u \in [0, L_i - 1], j = 1; \\ \frac{u}{(j-1)}, u \in [0, j-1], j = \overline{2, L_i - 1}; \\ \frac{(L_i - 1 - u)}{(L_i - j)}, u \in [j-1, L_i - 1], j = \overline{2, L_i - 1}; \\ \frac{u}{(L_i - 1)}, u \in [0, L_i - 1], j = L_i; \end{cases};$$

Найдем аналитическое выражение термов ($L_i = 7$), графическое изображение которых показано (см. рис.1-12)

$$u_i^* = \frac{6(x_i^* - x_{hi})}{(x_{bi} - x_{hi})};$$

$$\begin{aligned} \mu_1(u) &= 1 - \frac{u}{6}, \quad u \in [0, 6]; & \mu_2(u) &= \begin{cases} u, & u \in [0, 1] \\ \frac{(6-u)}{5}, & u \in [3, 6] \end{cases}; \\ \mu_3(u) &= \begin{cases} \frac{u}{2}, & u \in [0, 2] \\ \frac{(6-u)}{4}, & u \in [2, 6] \end{cases}; & \mu_4(u) &= \begin{cases} \frac{u}{3}, & u \in [0, 3] \\ \frac{(6-u)}{3}, & u \in [3, 6] \end{cases}; \\ \mu_5(u) &= \begin{cases} \frac{u}{4}, & u \in [0, 4] \\ \frac{(6-u)}{2}, & u \in [4, 6] \end{cases}; & \mu_6(u) &= \begin{cases} \frac{u}{5}, & u \in [0, 5] \\ 6-u, & u \in [5, 6] \end{cases}; \end{aligned}$$

$$\mu_7(u) = \frac{u}{6}, \quad u \in [0,6].$$

Полученные выражения могут быть использованы для выполнения задач принятия решения.

Наиболее часто используются две треугольные, симметричные относительно абсциссы $u = 0,5$ на едином универсуме $U = \{0,1\}$, функции принадлежности в виде:

$$\mu_1(u) = (1-u), \quad \mu_2(u) = u, \quad 0 \leq u \leq 1.$$

Фаззификация на базе замены терм-множества сигмоидных функций принадлежности сетью из искусственных нейронов

Построение нелинейных форм функций принадлежности (колоколообразная, функция Гаусса, сигмоидные и т.д.) осуществляется с помощью сети на искусственных нейронах на основании их внешней сходимости.

Рассмотрим реализацию терм-множества фаззификатора для лингвистической переменной «температура» с применением искусственной нейронной сети (ИНС). Под x понимается, например, отклонение температуры. Процесс фаззификации, в отличие от синглетонной базы, выполняется с помощью ИНС. Пусть терм-множество состоит из пяти терм: отрицательная средняя (ОС), отрицательная малая (ОМ), норма (Н), положительная малая (ПМ), положительная средняя (ПС). ИНС включает в себя восемь искусственных нейронов с сигмоидальными функциями активации и показана на рис. 1.13, где выходы y_1, y_2, y_3, y_4, y_5 есть термы фаззификатора: ОС, ОМ, Н, ПМ, ПС. Узлы со знаком + суммируют сигналы входов нейронов, а узлы с символом f реализуют их сигмоидальные функции. Сеть содержит дополнительно три нейрона с единичной активационной функцией (сумматор).

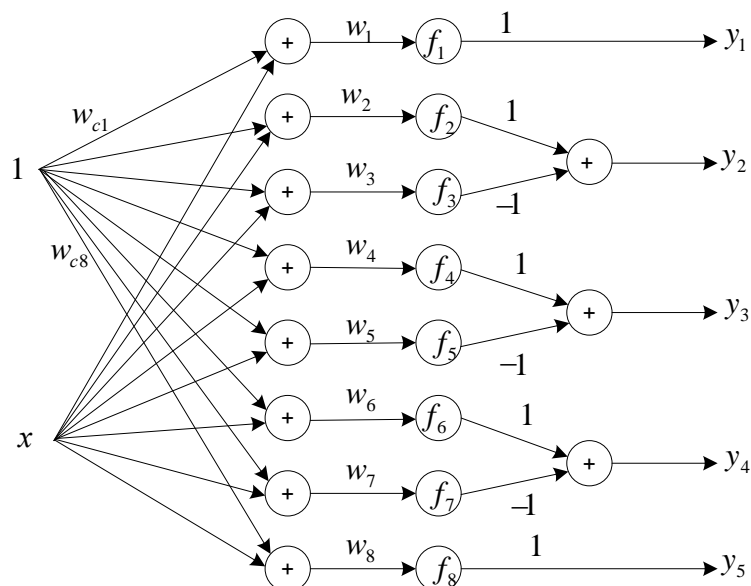


Рис.1-13. Нейросетевая реализация пяти терм фаззификатора
Выходы стандартной НС определяется по формулам:

$$y_1 = \text{OC} = \frac{1}{1 + \exp(-w_1(x + w_{c1}))};$$

$$y_2 = \text{OM} = \frac{1}{1 + \exp(-w_2(x + w_{c2}))} - \frac{1}{1 + \exp(-w_3(x - w_{c3}))};$$

$$y_3 = \text{H} = \frac{1}{1 + \exp(-w_4(x + w_{c4}))} - \frac{1}{1 + \exp(-w_5(x - w_{c5}))};$$

$$y_4 = \text{ПМ} = \frac{1}{1 + \exp(-w_6(x + w_{c6}))} - \frac{1}{1 + \exp(-w_7(x - w_{c7}))};$$

$$y_5 = \text{ПС} = \frac{1}{1 + \exp(-w_8(x - w_{c8}))};$$

где w_{c_i} – параметры смещения сигмоидальных ФП;

w_i – вес суммарного сигнала на входе сигмоидальных функций.

Выходы сети есть степени принадлежности активизированных терм. Если сигнала входа нет, то термы фаззификатора неактивизированы и проекции выходного вектора равны нулю или (нечеткое множество выхода фаззификатора равно нулю). Аналогично строится нейросетевой модуль для любого терм-множества фаззификатора. Путем подстройки весовых коэффициентов НС формирует функции принадлежности (термы) фаззификатора.

1.5. Расширение четкой логики И, ИЛИ, НЕ

Рассмотрим расширение четких логических операций И, ИЛИ, НЕ до нечетких операций: T -норма или S -конорма, S -норма или T -конорма, и дополнение. В нечетком пространстве число состояний неограниченно велико, поэтому невозможно описать эти операции с помощью таблиц истинности, как в случае двузначной логики.

Логика И

Один из подходов к оператору пересечения является его определение в классе треугольных T -норм и S -конорм.

Четкий элемент И: $y = x_1 \& x_2$ (бинарная форма); $y = x_1 \wedge x_2$.

Треугольной нормой (T -нормой) (расширенной связкой И называется двуместная действительная функция $T:[0,1] \times [0,1] \rightarrow [0,1]$, удовлетворяющая следующим условиям

1. $T(0,0) = 0$; $T(\mu_A, 1) = \mu_A$; $T(1, \mu_A) = \mu_A$ – граничное условие;
2. $T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D)$, если $\mu_A \leq \mu_C$, $\mu_B \leq \mu_D$ – монотонность;
3. $T(\mu_A, \mu_B) = T(\mu_B, \mu_A)$ – коммутативность;
4. $T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C)$ – ассоциативность;

Простым случаем треугольных норм являются:

- а) $\min(\mu_A, \mu_B)$ (пересечение по Заде);
- б) произведение $\mu_A \times \mu_B$ (умножение по Ларсену);
- в) $\max(0, \mu_A + \mu_B - 1)$ (пересечение по Лукашевичу)

Это действительная функция двух переменных, принимающая значение в единичном интервале $[0,1]$ может быть описана следующими выражениями:

$$\mu_{A_2}(x) = \mu_{A_1 \cap A_2}(x) = \mu_{A_1} \cap \mu_{A_2} = \mu_{A_1}(x) \cap \mu_{A_2}(x) = \mu_{A_1}(x)^T \cdot \mu_{A_2}(x) = T(\mu_{A_1}(x), \mu_{A_2}(x)) = \min\{\mu_{A_1}, \mu_{A_2}\}$$

где $\mu_{A_1}(x) \in [0,1]$; $\mu_{A_2}(x) \in [0,1]$; $\mu_{A_3}(x) \in [0,1]$;

Графическая интерпретация расширенного элемента «И» показана на рис. 1-14.

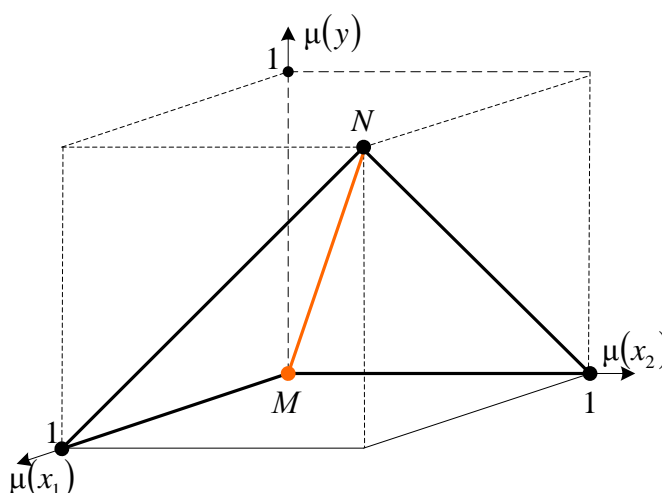


Рис. 1-14. Расширенный элемент И (Т-норма):

Линия MN – линия пересечения входных функций принадлежности

Логика ИЛИ

Один из подходов к оператору объединение является его определение в классе треугольных S -норм и T -конорм.

Четкий элемент *ИЛИ*: $y = x_1 \vee x_2$ (бинарная форма).

Треугольной нормой (S -нормой) (расширенная связка *ИЛИ*) называется двуместная действительная функция $S: [0,1] \times [0,1] \rightarrow [0,1]$, со свойствами:

1. $S(1,1) = 1$; $S(\mu_A, 0) = \mu_A$; $S(0, \mu_A) = \mu_A$ – ограниченность;
2. $S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D)$, если $\mu_A \geq \mu_C$, $\mu_B \geq \mu_D$ – монотонность;
3. $S(\mu_A, \mu_B) = S(\mu_B, \mu_A)$ – коммутативность;
4. $S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C)$ – ассоциативность.

Варианты S -норм:

- а) $\max(\mu_A, \mu_B)$ (объединение Заде);
- б) $\mu_A + \mu_B - \mu_A \times \mu_B$ (вероятностное ИЛИ);
- с) $\min(1, \mu_A + \mu_B)$ (объединение по Лукашевичу).

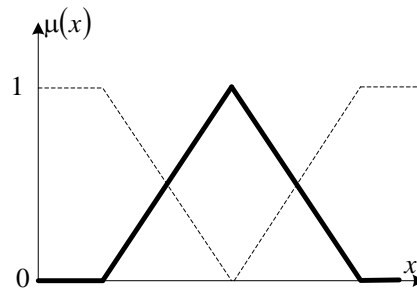


Рис.1-16. Дополнение нечеткого множества

Другие варианты реализации операций пересечение и объединение.

Группа логических операций (*max – min*):

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\};$$

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}.$$

Группа алгебраических операций:

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x);$$

$$\mu_{A \odot B}(x) = \mu_A(x) \cdot \mu_B(x).$$

Группа ограничений:

$$\mu_{A \oplus B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\};$$

$$\mu_{A \odot B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}.$$

Связь между *T*-нормы и *S*-нормы или расширенными связками *И* и *ИЛИ*

$$a \stackrel{T}{b} = 1 - \left[(1-a) \stackrel{S}{(1-b)} \right] = a + b - a \cdot b.$$

Широкое использование *T*- и *S*-норм обусловлено возможностью настройки параметров этих нечетких операторов, а также их хорошими алгебраическими свойствами, необходимыми для решения прикладных задач анализа и моделирования различных систем.

1.6. Нечеткие отношения. Операции с нечеткими множествами.

Декартово произведение

Нечеткие отношения необходимы для организации нечетких выводов [6]. Предположим, что знание эксперта отражает нечеткое причинное отношение предпосылки *A* и заключения *B*, которое описывается нечетким *R*.

$$R = A \rightarrow B,$$

где R – сила связи между элементами предпосылки A и заключения B .

Почти все реально работающие прикладные системы, использующие промежуточные нечеткие оценки, это системы, основанные на нечетких продукционных правилах. Нечеткое отношение R между двумя подмножествами X и Y будем называть нечетким подмножеством, определенным на декартовом произведении $X*Y$.

Пусть A_1, A_2, \dots, A_n – нечеткие подмножества универсальных множеств E_1, E_2, \dots, E_n . Тогда $A = A_1 \times A_2 \times \dots \times A_n$ – есть произведение подмножеств, лежащих в универсуме $E = E_1 \times E_2 \times \dots \times E_n$ с функцией принадлежности

$$\mu_A(x_1, x_2, \dots, x_n) = \min \{ \mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n) \}.$$

Формы записи нечеткого отношения:

$$R \in X*Y = \{(xRy): x \in X, y \in Y\} = \{(x, y), \mu_R(x, y)\}, \text{ где } \mu_R(x, y) \in [0,1].$$

$$R = \sum_{x \times y} \frac{\mu_R(x, y)}{(x, y)}.$$

В случае, когда $X = Y$, т.е. X и Y совпадают, нечеткое отношение $R:(X*X) \rightarrow [0,1]$ называется нечетким отношением на подмножестве X .

Пусть $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$.

Нечеткое отношение xRy может быть задано с помощью табл. 1-5.

Таблица 1-5 Нечеткое отношение

	y_1	y_2	y_3
x_1	0,8	0,6	0,4
x_2	1	0,8	0,6
x_3	0,8	1	0,8

Другие формы представления отношений

Пример 1-12. Найти отношение подмножеств $X = \{3, 4, 5\}$ и $Y = \{4, 5, 6\}$, где $x_1 = 3; x_2 = 4; x_3 = 5; y_1 = 4; y_2 = 5; y_3 = 6$. Их отношение: «у примерно равен х».

Решение. Данное отношение можно записать в виде

$$R = \frac{1}{(4,4)} + \frac{1}{(5,5)} + \frac{0,8}{(3,4)} + \frac{0,8}{(4,5)} + \frac{0,8}{(5,6)} + \frac{0,6}{(3,5)} + \frac{0,6}{(4,6)} + \frac{0,4}{(3,6)}$$

Или

$$\mu_R(x,y) = \begin{cases} 1, & \text{если } x = y; \\ 0,8, & \text{если } |x - y| = 1; \\ 0,6, & \text{если } |x - y| = 2; \\ 0,4, & \text{если } |x - y| = 3. \end{cases}$$

Или в виде матрицы

$$\mu_R(x, y) = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0,8 & 0,6 & 0,4 \\ x_2 & 1 & 0,8 & 0,6 \\ x_3 & 0,8 & 1 & 0,8 \end{array},$$

Пример 1-13. Определить нечеткое отношение с помощью операции *min* при заданных нечетких подмножествах *A* и *B*.

$$A = \left(\frac{0}{1}; \frac{0.1}{2}; \frac{0.5}{3}; \frac{0.8}{4}; \frac{1}{5}\right); \quad B = \left(\frac{1}{5}; \frac{0.8}{10}; \frac{0.4}{15}; \frac{0.2}{20}\right)$$

Решение

$$R = \begin{array}{c|cccc} & 0 & 0 & 0 & 0 \\ \hline & 0.1 & 0.1 & 0.1 & 0.1 \\ & 0.5 & 0.5 & 0.4 & 0.2 \\ & 0.8 & 0.8 & 0.4 & 0.2 \\ & 1 & 0.8 & 0.4 & 0.2 \end{array}$$

Операции с нечеткими отношениями

Пересечение

Пересечение двух отношений R_1 и R_2 обозначается $R_1 \cap R_2$ и определяется выражением $\mu_{R_1 \cap R_2}(x, y) = \mu_{R_1}(x, y) \cap \mu_{R_2}(x, y)$.

Объединение

Объединение двух отношений обозначается $R_1 \cup R_2$ и определяется выражением $\mu_{R_1 \cup R_2}(x, y) = \mu_{R_1}(x, y) \cup \mu_{R_2}(x, y)$.

Алгебраическое произведение

Алгебраическое произведение двух отношений R_1 и R_2 обозначается $R_1 * R_2$ и определяется выражением $\mu_{R_1 * R_2}(x, y) = \mu_{R_1}(x, y) * \mu_{R_2}(x, y)$.

Алгебраическая сумма

Алгебраическая сумма двух отношений R_1 и R_2 обозначается $R_1 \dot{+} R_2$ с выражением $\mu_{R_1 \dot{+} R_2}(x, y) = \mu_{R_1}(x, y) + \mu_{R_2}(x, y) - \mu_{R_1}(x, y) * \mu_{R_2}(x, y)$

Дизъюнктивная сумма

Дизъюнктивная сумма двух отношений R_1 и R_2 обозначается $R_1 \oplus R_2$ и определяется выражением $R_1 \oplus R_2 = (R_1 \cap \bar{R}_2) \cup (\bar{R}_1 \cap R_2)$

Дополнение

Дополнение отношения R обозначается \bar{R} и определяется функцией

Принадлежности $\mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y)$

Для введенных операций справедливы следующие свойства дистрибутивности

$$R_1 \cap (R_2 \cup R_3) = (R_1 \cap R_2) \cup (R_1 \cap R_3),$$

$$R_1 \cup (R_2 \cap R_3) = (R_1 \cup R_2) \cap (R_1 \cup R_3),$$

$$R_1 * (R_2 \cup R_3) = (R_1 * R_2) \cup (R_1 * R_3),$$

$$R_1 * (R_2 \cap R_3) = (R_1 * R_2) \cap (R_1 * R_3),$$

$$R_1 \dot{+} (R_2 \cup R_3) = (R_1 \dot{+} R_2) \cup (R_1 \dot{+} R_3),$$

$$R_1 \dot{+} (R_2 \cap R_3) = (R_1 \dot{+} R_2) \cap (R_1 \dot{+} R_3),$$

Рассмотрим возможные варианты комбинаций

1. Пересечение и объединение нечетких отношений

$$\mu_{R_1 \cap R_2}(x, z) = \min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))$$

$$\mu_{R_1 \cup R_2}(x, z) = \max(\mu_{R_1}(x, y), \mu_{R_2}(y, z))$$

2. Произведение нечетких отношений рассмотрим на примере 1-14.

Пример 1-14. Пусть заданы отношения

$$\begin{array}{c}
 y_1 \quad y_2 \qquad \qquad z_1 \quad z_2 \quad z_3 \\
 R_1 = \begin{matrix} x_1 \\ x_2 \end{matrix} \begin{bmatrix} 0,2 & 0,5 \\ 0,6 & 1 \end{bmatrix}; \quad R_2 = \begin{matrix} y_1 \\ y_2 \end{matrix} \begin{bmatrix} 0,3 & 0,6 & 0,8 \\ 0,7 & 0,9 & 0,4 \end{bmatrix},
 \end{array}$$

причем

$$X = \{x_1, x_2\}; Y = \{y_1, y_2\}; Z = \{z_1, z_2, z_3\}.$$

Решение

$$\begin{array}{c}
 y_1 \quad y_2 \qquad \qquad z_1 \quad z_2 \quad z_3 \qquad q_{11} \quad q_{12} \quad q_{13} \\
 R = R_1 * R_2 = \begin{matrix} x_1 \\ x_2 \end{matrix} \begin{bmatrix} 0,2 & 0,5 \\ 0,6 & 1 \end{bmatrix} * \begin{matrix} y_1 \\ y_2 \end{matrix} \begin{bmatrix} 0,3 & 0,6 & 0,8 \\ 0,7 & 0,9 & 0,4 \end{bmatrix} = \begin{bmatrix} 0,5 & 0,5 & 0,4 \\ 0,7 & 0,9 & 0,4 \end{bmatrix} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad q_{21} \quad q_{22} \quad q_{23}
 \end{array}$$

где

$$\begin{aligned}
 q_{11} &= \max[\min(0,2;0,3), \min(0,5;0,7)] = 0,5; \\
 q_{12} &= \max[\min(0,2;0,6), \min(0,5;0,9)] = 0,5; \\
 q_{13} &= \max[\min(0,2;0,8), \min(0,5;0,4)] = 0,4; \\
 q_{21} &= \max[\min(0,6;0,3), \min(1;0,7)] = 0,7; \\
 q_{22} &= \max[\min(0,6;0,6), \min(1;0,9)] = 0,9; \\
 q_{23} &= \max[\min(0,6;0,8), \min(1;0,4)] = 0,6.
 \end{aligned}$$

Пример 1-15. Рассмотрим нейросетевую реализацию нечетких отношений [5].

Основные трудности построения нечетких отношений связаны с настройкой большого числа функций принадлежности. Применение нейронной сети с синапсами, соответствующими этим функциям принадлежности, снимает эту проблему.

Пусть X и Y нечеткие множества на универсумах U и V , соответственно, и R , описываемых всеми отношениями (связями) между входом и выходом в декартовом произведении $U \times V$.

Запишем уравнение нечеткого отношения

$$Y = X \circ R.$$

Пусть $U = \{x_1, x_2, \dots, x_n\}$ и $V = \{y_1, y_2, \dots, y_n\}$.

Тогда следует

$$\mu_Y(y_j) = \max_{x_i} (\min(\mu_X(x_i), \mu_R(x_i, y_j))). \quad (1.2)$$

Выражение (1.2) можно записать в виде (1.3), используя матричную запись отношения R :

$$X \circ R = [\mu_X(x_1), \mu_X(x_2), \dots, \mu_X(x_n)] \circ \begin{pmatrix} \mu_R(x_1, y_1) & \dots & \mu_R(x_1, y_m) \\ \mu_R(x_2, y_1) & \dots & \mu_R(x_2, y_m) \\ \dots & \dots & \dots \\ \mu_R(x_n, y_1) & \dots & \mu_R(x_n, y_m) \end{pmatrix}. \quad (1.3)$$

Данное выражение может быть реализовано нейронной сетью на нечетких нейронах, показанной на рис. 3-7.

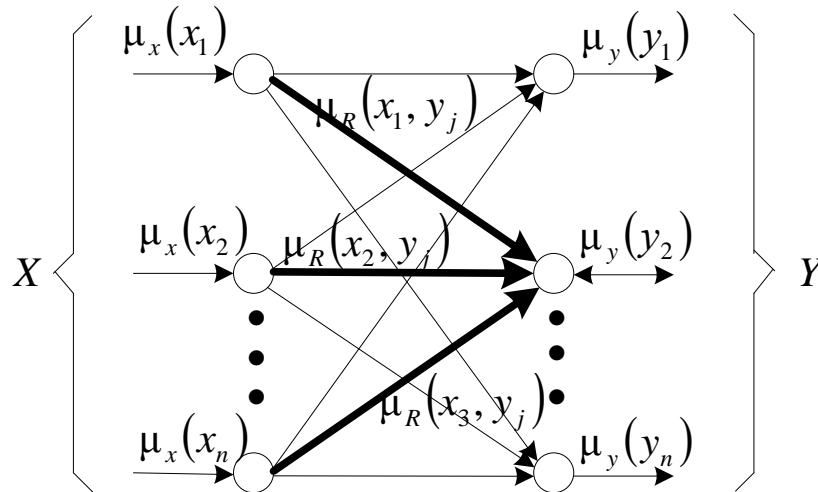


Рис. 1-17. Нейросетевая модель нечетких отношений

Однако, следуя уравнению (3.2), на определенном выходном узле необходимо получить сигнал (\max или \min), отличающийся от сигналов на других узлах. Если использовать операцию суммирования сигналов на входах нейрона, то результирующий сигнал может быть слишком малым для активационного выхода. Поэтому к каждому выходному узлу рассматриваемой нейронной сети добавляется порог (*bias*), обозначаемый $b_j \in [0,1]$.

Тогда уравнение (1.4) имеет вид:

$$\mu_Y(y_j) = \max_{x_j} [\max(\min(\mu_X(x_i), \mu_R(x_i, y)), b(y))]. \quad (1.4)$$

Добавление порога (точно так же, как в искусственном нейроне) можно рассматриваться как введение дополнительного входа в нейронный узел, значение которого изменяется от 0 до 1.

Таким образом, уравнение (1.4) можно рассматривать как функцию активации, используемую в выходном узле. Из этого уравнения также следует, что возможный уровень сигнала, получаемого на выходном узле сети, не может быть ниже порога. Для обучения такой сети можно использовать стандартные алгоритмы обучения.

1.7. Нечеткая импликация. Варианты реализации

Нечеткая импликация есть логическая операция отношения двух высказываний A и B в новое высказывание “Если A , то B ” и обозначается $R_{A \rightarrow B}$.

В более общем смысле нечеткую импликацию можно представить в виде нечетких подмножеств с функцией принадлежности, заданной выражением [5]

$$R_{A^n \rightarrow B^k} : IF (x_1 \text{ это } A_1^k \text{ AND } \dots \text{ AND } x_n \text{ это } A_n^k) THEN (y \text{ это } B^k).$$

Аналитическое выражение нечеткой импликации:

$$\int_{X*Y} \frac{\mu_R(x, y)}{\mu(x, y)} - \text{называется аналоговой нечеткой импликацией.}$$

$$\sum_{X*Y} \frac{\mu_R(x, y)}{\mu(x, y)} - \text{называется цифровой нечеткой импликацией}$$

Правило четкой импликации

1. Бинарное правило (Клинс), (1938г.) (Четкая импликация S -типа)

$$\mu_{R_{A \rightarrow B}}(x, y) = [1 - \mu_A(x)] \vee \mu_B(y) = \max[1 - \mu_A(x), \mu_B(y)].$$

Правила нечеткой импликации

1. Правило типа «логическое произведение», (1965г.) (Импликация T -типа по Заде) (правило Заде, расширение по Заде)

$$\mu_{R_{A \rightarrow B}} = [\mu_A(x) \cap \mu_B(y)] \cup [1 - \mu_A(x)] = \max \{ \min [\mu_A(x), \mu_B(y)], \mu_{\bar{A}}(x) \}$$

2. Алгоритм Мамдани (1974г.) (Импликация T -типа по Мамдани)
(правило Мамдани)

$$\mu_{R_{A \rightarrow B}}(x, y) = \mu_A(x) \cap \mu_B(y) = \min [\mu_A(x), \mu_B(y)].$$

1. Правило Лукашевича (1976г.) (правило Лукашевича)

$$\mu_{R_{A \rightarrow B}}(x, y) = 1 \cap [1 - \mu_A(x) + \mu_B(y)] = \min [1; 1 - \mu_A(x) + \mu_B(y)]$$

2. Правило типа «алгебраическое произведение» (Ларсен), (1980г.)
(правило Ларсена)

$$\mu_{R_{A \rightarrow B}}(x, y) = \mu_A(x) * \mu_B(y).$$

3. Правило Гогуэна нечеткая импликация типа

$$\mu_{R_{A \rightarrow B}}(x, y) = \begin{cases} 1, \text{ если } \mu_A(x) = 0 \\ \min\left(\frac{\mu_B(y)}{\mu_A(x)}; 1\right), \text{ для других} \end{cases}$$

4. Правило Шарпа

$$\mu_{R_{A \rightarrow B}}(x, y) = \begin{cases} 1, \text{ если } \mu_A(x) \leq \mu_B(y); \\ 0, \text{ если } \mu_A(x) > \mu_B(y) \end{cases}$$

5. Правило Гёделя

$$\mu_{R_{A \rightarrow B}}(x, y) = \begin{cases} 1, \text{ если } \mu_A(x) \leq \mu_B(y); \\ \mu_B(y), \text{ если } \mu_A(x) > \mu_B(y). \end{cases}$$

6. Вероятностное правило. (Рейшенбах). Импликация QL -типа.

$$\mu_{R_{A \rightarrow B}}(x, y) = 1 \cap [1 - \mu_A(x) + \mu_A(x)\mu_B(y)] = \min [1, 1 - \mu_A(x) + \mu_A(x)\mu_B(y)]$$

Операцию «нечеткой импликация» в теории нечетких множеств можно реализовать поразному: (при этом будет отличаться и полученной результат): по Заде, Мамдани, Ларсену и т.д.

1.8. Нечеткая композиция. Аналитический способ свертки

Нечеткая композиция есть свертка логической информации после нечеткой импликации [5].

Существует два способа композиции (свертки): аналитический и графический.

Аналитический способ композиции

Пусть $R_1: (X \times Y) \rightarrow [0,1]$; $R_2: (Y \times Z) \rightarrow [0,1]$

$\mu_{R_1 \cdot R_2}(x,z) = \cup [\mu_{R_1}(x, y) \cap \mu_{R_2}(y, z)]$ – есть *max-min* композиция (свертка) отношений R_1 и R_2 .

Пример 1-16. Определить композицию

R_1	y_1	y_2	Y_3
X_1	0.1	0.7	0.4
X_2	1	0.5	0

R_2	Z_1	Z_2	z_3	z_4
y_1	0.9	0	1	0.2
y_2	0.3	0.6	0	0.9
y_3	0.1	1	0	0.5

Тогда

$R_1 \cdot R_2$	z_1	Z_2	z_3	Z_4
x_1	0.3	0.6	0.1	0.7
x_2	0.9	0.5	1	0.5

$x_2 = y_1 = z_3$ – одни и те же числа.

Доказательство:

$$\mu_{R_1 \cdot R_2}(x_1, z_1) = [\mu_{R_1}(x_1, y_1) \cap \mu_{R_2}(y_1, z_1)] \cup [\mu_{R_1}(x_1, y_2) \cap \mu_{R_2}(y_2, z_1)] \cup$$

$$[\mu_{R1}(x_1, y_3) \cap \mu_{R2}(y_3, z_1)] = (0.1 \cap 0.9) \cup (0.7 \cap 0.3) \cup (0.4 \cap 0.1) = 0.1 \cup 0.3 \cup 0.1 = 0.3;$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_1, z_2) &= [\mu_{R1}(x_1, y_1) \cap \mu_{R2}(y_1, z_2)] \cup [\mu_{R1}(x_1, y_2) \cap \mu_{R2}(y_2, z_2)] \cup \\ &[\mu_{R1}(x_1, y_3) \cap \mu_{R2}(y_3, z_2)] = (0.1 \cap 0) \cup (0.7 \cap 0.6) \cup (0.4 \cap 1) = 0 \cup 0.6 \cup 0.4 = 0.6; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_1, z_3) &= [\mu_{R1}(x_1, y_1) \cap \mu_{R2}(y_1, z_3)] \cup [\mu_{R1}(x_1, y_2) \cap \mu_{R2}(y_2, z_3)] \cup \\ &[\mu_{R1}(x_1, y_3) \cap \mu_{R2}(y_3, z_3)] = (0.1 \cap 1) \cup (0.7 \cap 0) \cup (0.4 \cap 0) = 0.1 \cup 0 \cup 0 = 0.1; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_1, z_4) &= [\mu_{R1}(x_1, y_1) \cap \mu_{R2}(y_1, z_4)] \cup [\mu_{R1}(x_1, y_2) \cap \mu_{R2}(y_2, z_4)] \cup [\mu_{R1}(x_1, y_3) \\ &\cap \mu_{R2}(y_3, z_4)] = (0.1 \cap 0.2) \cup (0.7 \cap 0.9) \cup (0.4 \cap 0.5) = 0.1 \cup 0.7 \cup 0.4 = 0.7; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_2, z_1) &= [\mu_{R1}(x_2, y_1) \cap \mu_{R2}(y_1, z_1)] \cup [\mu_{R1}(x_2, y_2) \cap \mu_{R2}(y_2, z_1)] \cup [\mu_{R1}(x_2, y_3) \\ &\cap \mu_{R2}(y_3, z_1)] = (1 \cap 0.9) \cup (0.5 \cap 0.3) \cup (0 \cap 0.1) = 0.9 \cup 0.3 \cup 0 = 0.9; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_2, z_2) &= [\mu_{R1}(x_2, y_1) \cap \mu_{R2}(y_1, z_2)] \cup [\mu_{R1}(x_2, y_2) \cap \mu_{R2}(y_2, z_2)] \cup \\ &[\mu_{R1}(x_2, y_3) \cap \mu_{R2}(y_3, z_2)] = (1 \cap 0) \cup (0.5 \cap 0.6) \cup (0 \cap 1) = 0 \cup 0.5 \cup 0 = 0.5; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_2, z_3) &= [\mu_{R1}(x_2, y_1) \cap \mu_{R2}(y_1, z_3)] \cup [\mu_{R1}(x_2, y_2) \cap \mu_{R2}(y_2, z_3)] \cup [\mu_{R1}(x_2, y_3) \\ &\cap \mu_{R2}(y_3, z_3)] = (1 \cap 1) \cup (0.5 \cap 0) \cup (0 \cap 0) = 1 \cup 0 \cup 0 = 1; \end{aligned}$$

$$\begin{aligned} \mu_{R1 \cdot R2}(x_2, z_4) &= [\mu_{R1}(x_2, y_1) \cap \mu_{R2}(y_1, z_4)] \cup [\mu_{R1}(x_2, y_2) \cap \mu_{R2}(y_2, z_4)] \cup [\mu_{R1}(x_2, y_3) \\ &\cap \mu_{R2}(y_3, z_4)] = (1 \cap 0.2) \cup (0.5 \cap 0.9) \cup (0 \cap 0.5) = 0.2 \cup 0.5 \cup 0 = 0.5. \end{aligned}$$

Графический способ нечеткой композиции – поточечное суммирование ординат функций принадлежности после нечеткой импликации.

Свойства *max-min* нечеткой композиции

Операция *max-min* композиции – ассоциативна:

$$R_3 * (R_2 * R_1) = (R_3 * R_2) * R_1$$

дистрибутивна относительно объединения:

$$R_1 * (R_2 \cup R_3) = (R_1 * R_2) \cup (R_1 * R_3)$$

дистрибутивна относительно пересечения:

$$R_1 * (R_2 \cap R_3) = (R_1 * R_2) \cap (R_1 * R_3)$$

Кроме того, для *max-min* композиции выполняется следующее важное свойство (включение): если $R_1 \subset R_2$, то $R_3 * R_1 \subset R_3 * R_2$.

1.9. Модуль нечеткого логического вывода.

Варианты реализации

Для многих приложений, связанных с управлением технологических процессов, необходимо построение модели рассматриваемого процесса. Знание модели позволяет подобрать соответствующий регулятор. Применение теории нечетких подмножеств управления технологическими процессами не предполагает знание моделей этих процессов. Следует только сформулировать правила поведения в форме нечетких условных суждений.

Рассмотрим структуру модуля «нечеткий регулятор», показанного на рис. 1-18, состоящего из фаззификатора, блока нечеткого логического вывода и дефаззификатора.

Фаззификация нечеткого регулятора рассмотрена в разделе 1.4.

Блок нечеткого логического вывода объединяет операции нечеткой импликации и нечеткой композиции.

Дефаззификатор преобразует нечеткую логическую информацию в четкую логическую информацию.

Одним из вариантов нечеткого логического вывода является композиционное правило Заде

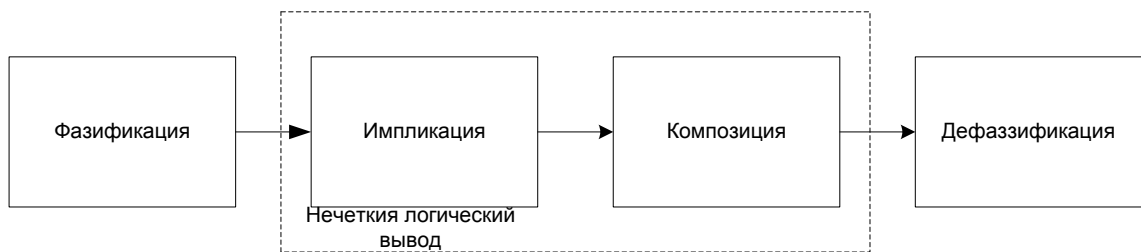


Рис. 1-18. Структурная схема модуля нечеткий регулятор

Композиционное правило нечеткого логического вывода Заде

Если известно нечеткое отношение переменных x и y , то при нечетком значении входной переменной $x = A$ нечеткое значение выходной переменной определяется так: $y = A \circ R$, где \circ – знак макс-мин композиции

Пример 1-17: Пусть заданы нечеткие подмножества

$$A = \left(\frac{0}{1} + \frac{0.1}{2} + \frac{0.5}{3} + \frac{0.8}{4} + \frac{1}{5} \right); \quad B = \left(\frac{1}{5} + \frac{0.8}{10} + \frac{0.4}{15} + \frac{0.2}{20} \right)$$

Определить значение выходной переменной y при

$$x = C = \left(\frac{0.3}{1} + \frac{0.5}{2} + \frac{1}{3} + \frac{0.7}{4} + \frac{0.4}{5} \right)$$

Вначале рассчитываем нечеткое отношение R согласно правилу: «Если $x = A$, то $y = B$ », применяя в качестве t -нормы операцию \min , получим:

$$R = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0.4 & 0.2 \\ 0.8 & 0.8 & 0.4 & 0.2 \\ 1 & 0.8 & 0.4 & 0.2 \end{vmatrix}$$

По формуле $y = C \circ R$ рассчитываем нечеткое значение функции принадлежности выходной переменной y

$$\begin{aligned} \mu_B(y) &= \sup_{x \in X} \{[\mu_A(x) \circ \mu_R(x, y)]\} = \sup_{x \in X} \{\min[\mu_A(x), \mu_R(x, y)]\} \\ &= \max_{x \in X} \{\min[\mu_A(x), \mu_R(x, y)]\} \\ \mu_B(y) &= \max_{x \in X} \min[\mu_A(x), \mu_{R_{A \rightarrow B}}(x, y)], \end{aligned}$$

т.е. функция принадлежности заключения B равна max значению функции принадлежности пересечения посылки A и импликации $A \rightarrow B$.

$$y = \left(\frac{0.3}{1} + \frac{0.5}{2} + \frac{1}{3} + \frac{0.7}{4} + \frac{0.4}{5} \right) \circ \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0.4 & 0.2 \\ 0.8 & 0.8 & 0.4 & 0.2 \\ 1 & 0.8 & 0.4 & 0.2 \end{vmatrix} = \left(\frac{0.7}{5} + \frac{0.7}{10} + \frac{0.4}{15} + \frac{0.2}{20} \right);$$

Доказательство:

$$\begin{aligned} q_1 &= \max(\min(0.3;0), \min(0.5;0.1) \min(1;0.5) \min(0.7;0.8) \min(0.4;1)) = 0.7; \\ q_2 &= \max(\min(0.3;0), \min(0.5;0.1) \min(1;0.5) \min(0.7;0.8) \min(0.4;0.8)) = 0.7; \\ q_3 &= \max(\min(0.3;0), \min(0.5;0.1) \min(1;0.4) \min(0.7;0.4) \min(0.4;0.4)) = 0.4; \\ q_4 &= \max(\min(0.3;0), \min(0.5;0.1) \min(1;0.2) \min(0.7;0.2) \min(0.4;0.2)) = 0.2; \end{aligned}$$

Композиционное правило Заде составляет основу разнообразных алгоритмов нечеткого логического вывода, где операцию «нечеткой импликация» в алгебре нечетких множеств можно реализовать поразному (при этом будет отличаться и полученной результат): Заде, Мамдани, Ларсену и т.д.

1.10. Графическая интерпретация системы нечеткого вывода

Рассмотрим наиболее важные варианты модификации нечеткого вывода, где алгоритмы нечеткого вывода различаются в основном видом используемого правила нечеткой импликации и терм-множеством функций принадлежности лингвистической переменной выхода нечеткого вывода.

Нечеткий вывод Заде

Модификации нечеткого вывода, где нечеткая импликация реализуется операцией взятия минимума и терм-множество управления задано синглетами [5, 6].

Пусть продуктивные правила Π_1, Π_2 представлены в следующей форме:

$$\Pi_1: \text{если } x \text{ есть } A_1 \text{ и } y \text{ есть } B_1, \text{ то } z \text{ есть } C_1;$$

$$\Pi_2: \text{если } x \text{ есть } A_2 \text{ и } y \text{ есть } B_2, \text{ то } z \text{ есть } C_2,$$

где x, y, z – имена переменных входа и выхода, соответственно, а $A_1, B_1, A_2, B_2, C_1, C_2$ – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение z_0 необходимо определить при входных переменных x_0 и y_0 .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил: $(\Pi_1, \Pi_2): A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$.

2. Находим степени принадлежности после операции минимум, где нечеткая импликация T -типа (по Мамдани) определяется по формуле:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)]$$

или

$$\alpha_1 = \min[A_1(x_0), B_1(y_0)] = A_1(x_0) \cap B_1(y_0);$$

$$\alpha_2 = \min[A_2(x_0), B_2(y_0)] = A_2(x_0) \cap B_2(y_0);$$

3. Находим степень принадлежности одноименных функций принадлежности после операции минимум, используя операцию максимум.

4. Определяем выход после нечеткой композиции

$$\sum_{i=1}^n \alpha_i C_i,$$

где функции принадлежности терм-множества лингвистической переменной выхода (синглетоны) C_1, C_2 задаются.

5. Методом центраида приводим к четкости переменную выхода

$$z_0 = \frac{\sum_{i=1}^n \alpha_i C_i}{\sum_{i=1}^n \alpha_i}$$

Графическая интерпретация нечеткого вывода показана на рис.1-19.

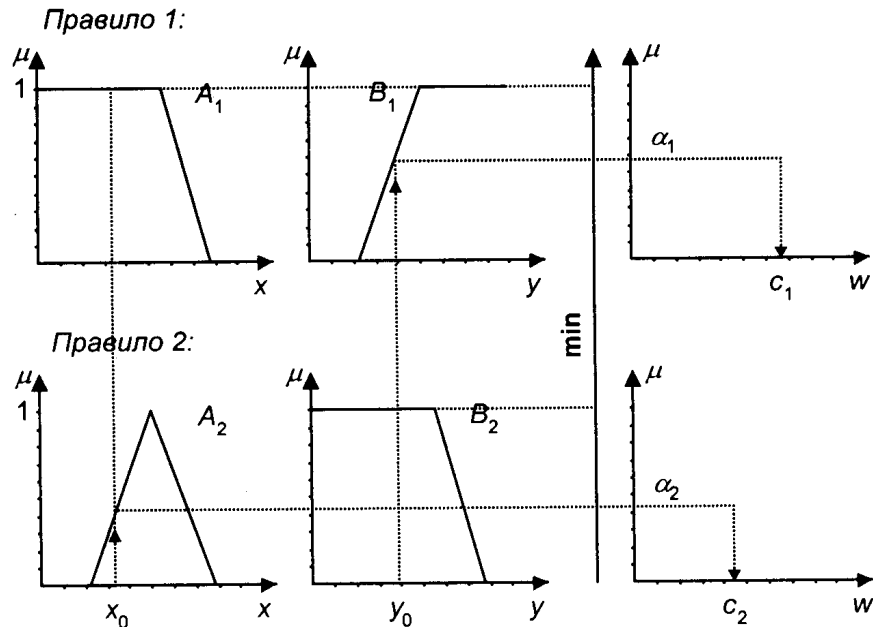


Рис.1-19. Графическая интерпретация нечеткого вывода по Заде

Нечеткий вывод Мамдани (Mamdani) (1974 г.)

В этом варианте нечеткого вывода используем фаззификацию на несинглетонной базе и операцию взятия минимума в качестве нечеткой импликации, показанного на рис.1-8 [6].

Пусть продуктивные правила Π_1, Π_2 записаны в следующей форме:

Π_1 : если x есть A_1 и y есть B_1 , то z есть C_1 ;

Π_2 : если x есть A_2 и y есть B_2 , то z есть C_2 ,

где x, y, z – имена переменных входа и выхода, соответственно, а $A_1, B_1, A_2, B_2, C_1, C_2$ – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение z_0 необходимо определить при текущих переменных x_0 и y_0 .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил: $(\Pi_1, \Pi_2): A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$.

2. Находим степени принадлежности после операции минимум, где нечеткая импликация T -типа (по Мамдани) определяется по формуле:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)]$$

или

$$\alpha_1 = \min[A_1(x_0), B_1(y_0)] = A_1(x_0) \cap B_1(y_0);$$

$$\alpha_2 = \min[A_2(x_0), B_2(y_0)] = A_2(x_0) \cap B_2(y_0);$$

3. Находятся «усеченные» функции принадлежности $C_1^*(z), C_2^*(z)$ для предпосылок каждого правила при нечеткой композиции:

$$C_1^* = [\alpha_1 \cap C_1] = \min\{\alpha_1, C_1\};$$

$$C_2^* = [\alpha_2 \cap C_2] = \min\{\alpha_2, C_2\}.$$

4. Нечеткая композиция (свертка) производится объединением найденных усеченных функций: \dot{C}_1, \dot{C}_2 с помощью поточечного суммирования:

$$C_{\Sigma}^* = C_1^* \cup C_2^*$$

5. Приведение к четкости модифицированным методом центраида.

$$x_{\text{ЦГ}}^{\Sigma} = \frac{1}{n} \sum_0^{n-1} (x_i + \frac{x_{i+1} - x_i}{2});$$

$$y_{\text{ЦГ}}^{\Sigma} = \frac{0,5}{n} \sum_0^{n-1} \frac{S_i}{x_{i+1} - x_i}.$$

Пример 1-18. Рассмотрим нечеткий вывод по Мамдани.

Пусть дана система управления нечеткой логики с двумя правилами нечеткого управления, где фаззификация выполнена на синглетонной базе:

Rule1: IF x is A₁ AND y is B₁ THEN z is C₁;

Rule2: IF x is A₂ AND y is B₂ THEN z is C₂

Предположим, что величины x_i и y_i , считываемые с датчиков, являются четкими входными величинами для лингвистических переменных x и y . При этом заданы следующие термы для нечетких подмножеств $A_1, A_2, B_1, B_2, C_1, C_2$ этих переменных:

$$\mu_{A_1}(x) = \begin{cases} \frac{(x-2)}{2} & \text{при } 2 \leq x \leq 4, \\ \frac{(6-x)}{2} & \text{при } 4 < x \leq 6. \end{cases}; \quad \mu_{A_2}(x) = \begin{cases} \frac{(x-1)}{3} & \text{при } 1 \leq x \leq 4, \\ \frac{(7-x)}{3} & \text{при } 4 < x \leq 7. \end{cases};$$

$$\mu_{B_1}(y) = \begin{cases} \frac{(y-0)}{3} & \text{при } 0 \leq y \leq 3, \\ \frac{(6-y)}{3} & \text{при } 3 < y \leq 6. \end{cases}; \quad \mu_{B_2}(y) = \begin{cases} \frac{(y-1)}{3} & \text{при } 1 \leq y \leq 4, \\ \frac{(7-y)}{3} & \text{при } 4 < y \leq 7. \end{cases};$$

$$\mu_{C_1}(z) = \begin{cases} \frac{(z-0)}{3} & \text{при } 0 \leq z \leq 3, \\ \frac{(6-z)}{3} & \text{при } 3 < z \leq 6. \end{cases}; \quad \mu_{C_2}(z) = \begin{cases} \frac{(z-0)}{4} & \text{при } 0 \leq z \leq 4, \\ \frac{(8-z)}{4} & \text{при } 4 < z \leq 8. \end{cases}.$$

Предположим, что в момент времени t_1 были считаны значения датчиков $x_0(t_1)=3$ и $y_0(t_1)=2$. Определяем срезы для обеих правил на основе заданных функций и с учетом $x_0(t_1)=3$ и $y_0(t_1)=2$.

$$\begin{aligned} \mu_{A_1}(x_0=3) &= 0,5; & \mu_{B_1}(y_0=2) &= 0,7; \\ \mu_{A_2}(x_0=3) &= 0,6; & \mu_{B_2}(y_0=2) &= 0,3. \end{aligned}$$

Затем в соответствии с правилом вывода Мамдани определяем уровни среза, показанных на рис.1-20.

$$\begin{aligned} \alpha_1 &= \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0)) = \min(0,5, 0,7) = 0,5; \\ \alpha_2 &= \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0)) = \min(0,6, 0,3) = 0,3. \end{aligned}$$

Окончательно, поточечно суммируя функции принадлежности выхода C'_1 и C'_2 , и используя формулу метода центраида, получим

$$z_{\text{цт}} = \frac{1 \cdot 0,5 + 2 \cdot 0,5 + 3 \cdot 0,5 + 4 \cdot 0,5 + 5 \cdot 0,3 + 6 \cdot 0,3 + 7 \cdot 0,3 + 8 \cdot 0}{0,5 \cdot 4 + 0,3 \cdot 3} = 4,98.$$

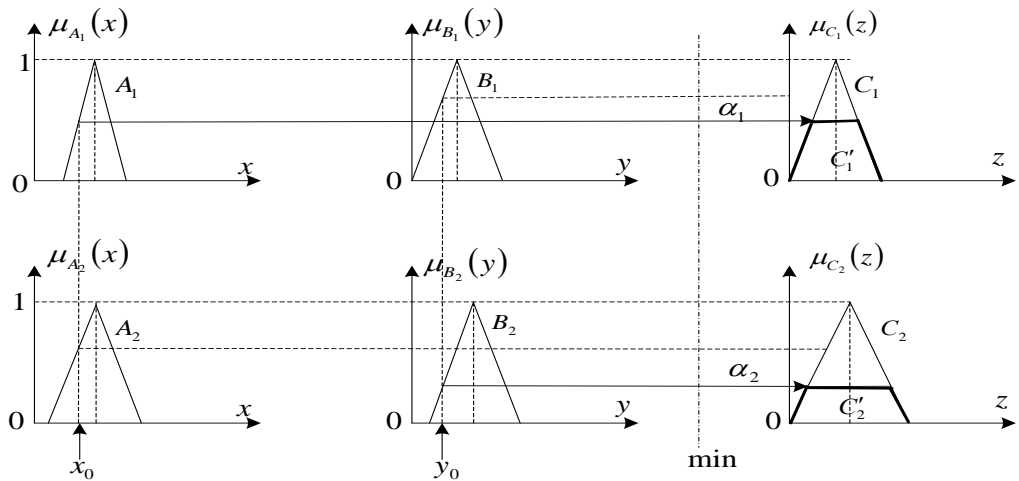


Рис.1-20. Иллюстрация нечеткого вывода по Мамдани

Нечеткий вывод по Ларсену (Larsen) (1980г.)

Пример 1-19. Рассмотрим нечеткий вывод по Ларсену [5].

Пусть дана система управления нечеткой логики с двумя правилами нечеткого управления:

Rule1: IF x is A₁ AND y is B₁ THEN z is C₁ ;

Rule2: IF x is A₂ AND y is B₂ THEN z is C₂ .

Предположим, что величины x_i и y_i , считываемые с датчиков, являются четкими входными величинами для лингвистических переменных x и y . При этом заданы следующие термы для нечетких подмножеств $A_1, A_2, B_1, B_2, C_1, C_2$ этих переменных:

$$\mu_{A_1}(x) = \left\{ \begin{array}{l} \frac{(x-2)}{2} \text{ при } 2 \leq x \leq 4, \\ \frac{(6-x)}{2} \text{ при } 4 < x \leq 6. \end{array} \right\}; \quad \mu_{A_2}(x) = \left\{ \begin{array}{l} \frac{(x-1)}{3} \text{ при } 1 \leq x \leq 4, \\ \frac{(7-x)}{3} \text{ при } 4 < x \leq 7. \end{array} \right\};$$

$$\mu_{B_1}(y) = \left\{ \begin{array}{l} \frac{(y-0)}{3} \text{ при } 0 \leq y \leq 3, \\ \frac{(6-y)}{3} \text{ при } 3 < y \leq 6. \end{array} \right\}; \quad \mu_{B_2}(y) = \left\{ \begin{array}{l} \frac{(y-1)}{3} \text{ при } 1 \leq y \leq 4, \\ \frac{(7-y)}{3} \text{ при } 4 < y \leq 7. \end{array} \right\};$$

$$\mu_{C_1}(z) = \begin{cases} \frac{(z-0)}{3} & \text{при } 0 \leq z \leq 3, \\ \frac{(6-z)}{3} & \text{при } 3 < z \leq 6. \end{cases}; \quad \mu_{C_2}(z) = \begin{cases} \frac{(z-0)}{4} & \text{при } 0 \leq z \leq 4, \\ \frac{(8-z)}{4} & \text{при } 4 < z \leq 8. \end{cases}.$$

Предположим, что в момент времени t_1 были считаны значения датчиков $x_0(t_1)=3$ и $y_0(t_1)=2$. Определяем срезы для обеих правил на основе заданных функций и с учетом $x_0(t_1)=3$ и $y_0(t_1)=2$.

$$\begin{aligned} \mu_{A_1}(x_0=3) &= 0,5; & \mu_{B_1}(y_0=2) &= 0,7; \\ \mu_{A_2}(x_0=3) &= 0,6; & \mu_{B_2}(y_0=2) &= 0,3. \end{aligned}$$

Затем в соответствии с правилом вывода алгоритма Мамдани определяем уровни среза (нечеткая импликация)

$$\begin{aligned} \alpha_1 &= \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0)) = \min(0,5, 0,7) = 0,5; \\ \alpha_2 &= \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0)) = \min(0,6, 0,3) = 0,3. \end{aligned}$$

Находим *усеченные* функции принадлежности $C^*_1(z), C^*_2(z)$ для предпосылок каждого правила при нечеткой композиции. В данном алгоритме нечеткая импликация моделируется с использованием оператора умножения:

$$\mu_{\dot{C}_1}(z) = \alpha_1 \begin{cases} \frac{(z-0)}{3} & \text{при } 0 \leq z \leq 3, \\ \frac{(6-z)}{3} & \text{при } 3 < z \leq 6. \end{cases}; \quad \mu_{\dot{C}_2}(z) = \alpha_2 \begin{cases} \frac{(z-0)}{4} & \text{при } 0 \leq z \leq 4, \\ \frac{(8-z)}{4} & \text{при } 4 < z \leq 8. \end{cases}$$

Или

$$\mu_{\dot{C}_1}(z) = \begin{cases} \frac{0,5(z-0)}{3} & \text{при } 0 \leq z \leq 3, \\ \frac{0,5(6-z)}{3} & \text{при } 3 < z \leq 6. \end{cases}; \quad \mu_{\dot{C}_2}(z) = \begin{cases} \frac{0,3(z-0)}{4} & \text{при } 0 \leq z \leq 4, \\ \frac{0,3(8-z)}{4} & \text{при } 4 < z \leq 8. \end{cases}.$$

Нечеткая композиция (свертка) производится объединением найденных усеченных функций: $\dot{C}_1(z), \dot{C}_2(z)$ с помощью поточечного суммирования:

$$\mu_{\Sigma}(z) = C(z) = C^*_1(z) \cup C^*_2(z).$$

Приведение к четкости модифицированным методом центраида.

$$x_{\text{ЦП}}^{\Sigma} = \frac{1}{n} \sum_0^{n-1} \left(x_i + \frac{x_{i+1} - x_i}{2} \right);$$

$$y_{\text{ЦП}}^{\Sigma} = \frac{0,5}{n} \sum_0^{n-1} \frac{S_i}{x_{i+1} - x_i}.$$

На рис.1-21 приведена графическая интерпретация нечеткого вывода по Ларсену.

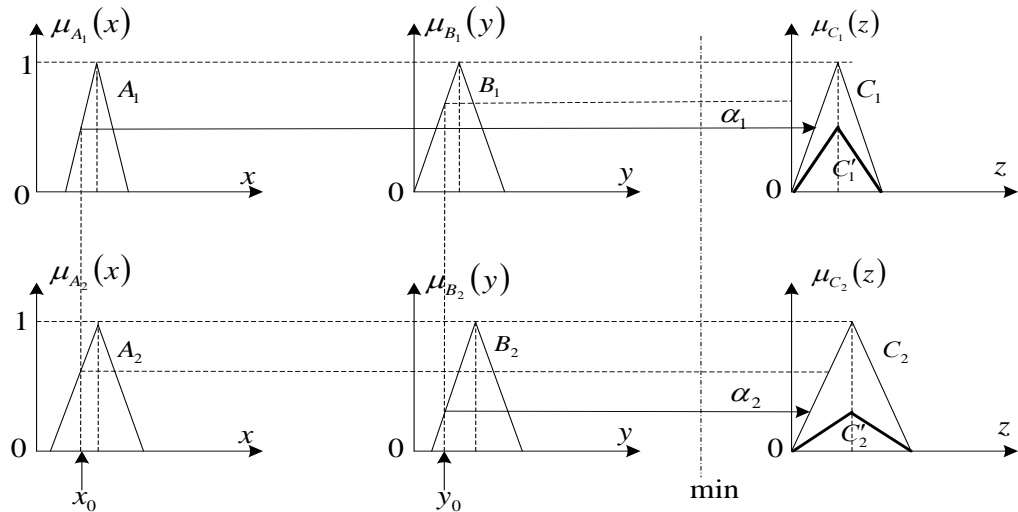


Рис. 1-21. Иллюстрация нечеткого вывода по Ларсену

Нечеткий вывод по Цукамото

В данном варианте нечеткого вывода используется тот же алгоритм, что и при выводе по Мамдани, но требуется, чтобы функции принадлежности выхода были монотонны (убывающие или возрастающие) [5].

Пусть продуктивные правила $\Pi_1 \dots \Pi_4$ записаны в следующей форме:

Π_1 : если x есть A_1 и y есть B_1 , то z есть C_1 ;

Π_2 : если x есть A_2 и y есть B_2 , то z есть C_2 ,

где x, y, z – имена переменных входа и выхода, соответственно, а $A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4, C_1, C_2$ – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение z_0 необходимо определить при текущих переменных x_0 и y_0 .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил: $(\Pi_1 \dots \Pi_4): A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$.

2. Находим степени принадлежности после операции минимум, где нечеткая импликация T -типа (по Мамдани) определяется по формуле:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)]$$

или

$$\alpha_1 = \min[A_1(x_0), B_1(y_0)] = A_1(x_0) \cap B_1(y_0);$$

$$\alpha_2 = \min[A_2(x_0), B_2(y_0)] = A_2(x_0) \cap B_2(y_0);$$

3. Определяем заключения с заменой аргумента на функция и наоборот

$$z_1 = \frac{1}{\mu_{C_1}(\alpha_1)} \qquad z_2 = \frac{1}{\mu_{C_2}(\alpha_2)}$$

4. Четкое значение выходной переменной определяется сразу как взвешенная комбинация

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}$$

Пример 1-20. Рассмотрим нечеткий вывод по Цукамото.

Пусть дана система управления нечеткой логики с двумя правилами нечеткого управления:

Rule1: IF x is A₁ AND y is B₁ THEN z is C₁;

Rule2: IF x is A₂ AND y is B₂ THEN z is C₂.

Предположим, что величины x_i и y_i , считываемые с датчиков, являются четкими входными величинами для лингвистических переменных x и y . При этом заданы следующие термы для нечетких подмножеств $A_1, A_2, B_1, B_2, C_1, C_2$ этих переменных:

$$\mu_{A_1}(x) = \left\{ \begin{array}{l} \frac{(x-2)}{2} \text{ при } 2 \leq x \leq 4, \\ \frac{(6-x)}{2} \text{ при } 4 < x \leq 6. \end{array} \right\}; \qquad \mu_{A_2}(x) = \left\{ \begin{array}{l} \frac{(x-1)}{3} \text{ при } 1 \leq x \leq 4, \\ \frac{(7-x)}{3} \text{ при } 4 < x \leq 7. \end{array} \right\};$$

$$\mu_{B_1}(y) = \begin{cases} \frac{(y-0)}{3} & \text{при } 0 \leq y \leq 3, \\ \frac{(6-y)}{3} & \text{при } 3 < y \leq 6. \end{cases}; \quad \mu_{B_2}(y) = \begin{cases} \frac{(y-1)}{3} & \text{при } 1 \leq y \leq 4, \\ \frac{(7-y)}{3} & \text{при } 4 < y \leq 7. \end{cases};$$

$$\mu_{C_1}(z) = 2z^2; \quad \mu_{C_2}(z) = 2\sqrt{z}.$$

На рис. 1-22 приведена иллюстрация нечеткого вывода по Цукamoto

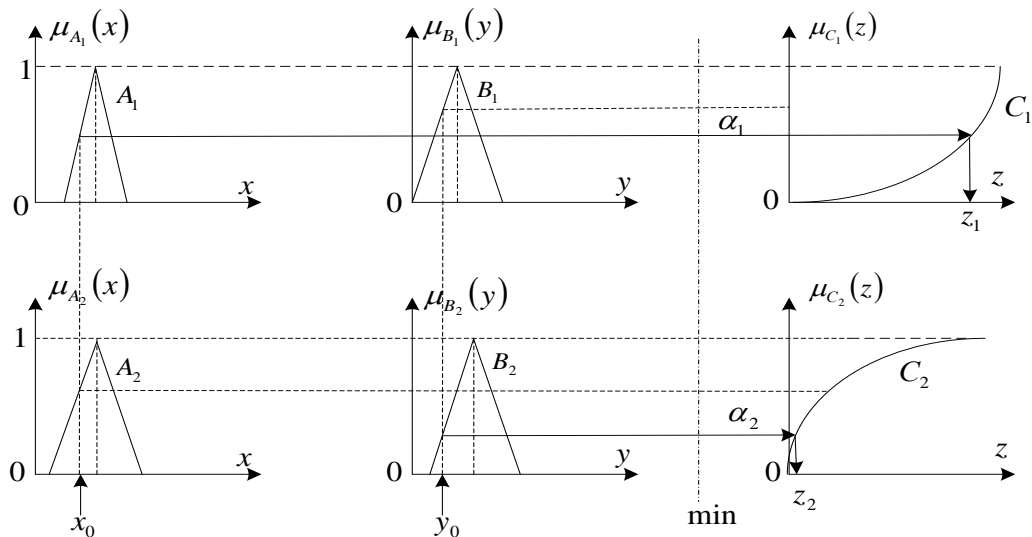


Рис. 1-22. Иллюстрация нечеткого вывода по Цукamoto

Предположим, что в момент времени t_1 были считаны значения датчиков $x_0(t_1)=3$ и $y_0(t_1)=2$. Определяем срезы для обеих правил на основе заданных функций и с учетом $x_0(t_1)=3$ и $y_0(t_1)=2$.

$$\mu_{A_1}(x_0 = 3) = 0,5; \quad \mu_{B_1}(y_0 = 2) = 0,7;$$

$$\mu_{A_2}(x_0 = 3) = 0,6; \quad \mu_{B_2}(y_0 = 2) = 0,3.$$

Затем в соответствии с правилом вывода алгоритма Мамдани определяем уровни среза (нечеткая импликация)

$$\alpha_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0)) = \min(0,5, 0,7) = 0,5;$$

$$\alpha_2 = \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0)) = \min(0,6, 0,3) = 0,3.$$

Определяем

$$z_1 = \frac{1}{\mu_{C_1}(\alpha_1)} = 12$$

$$z_2 = \frac{1}{\mu_{C_2}(\alpha_2)} = 0,5$$

Четкое значение выходной переменной определяется сразу как взвешенная комбинация

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} = \frac{0,5 \cdot 12 + 0,3 \cdot 0,5}{0,5 + 0,3} = 7,7$$

Нечеткий вывод по Сугено-Такаги

Пусть продуктивные правила $\Pi_1 \dots \Pi_4$ записаны в следующей форме:

Π_1 : если x есть A_1 и y есть B_1 , то $z_1 = a_1 x_1 + b_1 y_1$;

Π_2 : если x есть A_2 и y есть B_2 , то $z_2 = a_2 x_2 + b_2 y_2$;

где x, y, z – имена переменных входа и выхода, соответственно, а A_1, B_1, A_2, B_2 – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение z_0 необходимо определить через x_0 и y_0 .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил: ($\Pi_1 \dots \Pi_4$): $A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$.

2. Находим степени принадлежности после операции минимум, где нечеткая импликация T -типа (по Мамдани) определяется по формуле:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)]$$

или

$$\alpha_1 = \min[A_1(x_0), B_1(y_0)] = A_1(x_0) \cap B_1(y_0);$$

$$\alpha_2 = \min[A_2(x_0), B_2(y_0)] = A_2(x_0) \cap B_2(y_0);$$

3. Определяем «индивидуальные выходы»

$$z_1 = a_1 x + b_1 y;$$

$$z_2 = a_2 x + b_2 y,$$

где коэффициенты a_i и b_i задаются.

4. Методом центраида приводим к четкости переменной вывода:

$$z_0 = \frac{\sum_{i=1}^2 \alpha_i z_i}{\sum_{i=1}^2 \alpha_i} .$$

Пример 1-21. Пусть дана система управления модуля нечеткой логики с двумя правилами нечеткого управления:

Rule 1: IF \bar{x} is A_1^1 AND \bar{y} is A_2^1 THEN $z_1 = 4 + 2\bar{x} - 2\bar{y}$;

Rule 2: IF \bar{x} is A_1^2 AND \bar{y} is A_2^2 THEN $z_2 = 2 + 4\bar{x} - 2\bar{y}$;

Предположим, что величины \bar{x} и \bar{y} , считываемые с датчиков, являются четкими входными величинами. При этом заданы следующие термы нечетких подмножеств $A_1^1, A_2^1, A_1^2, A_2^2$ этих переменных.

Предположим, что в момент времени t были считаны значения $\bar{x}(t) = 4$ и $\bar{y}(t) = 5$.

Определим выходные сигналы для z $\bar{x}(t) = 4$ и $\bar{y}(t) = 5$ с учетом графиков на рис.1-23.

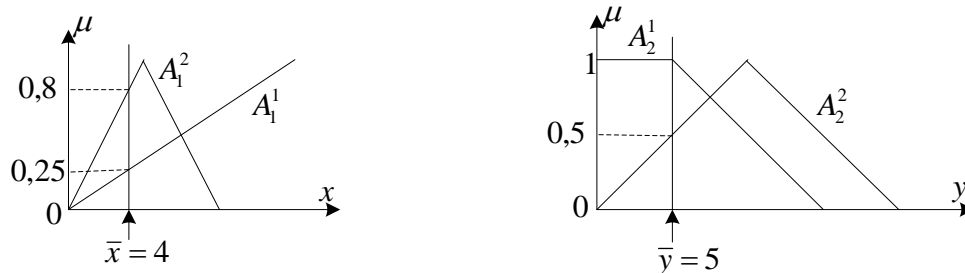


Рис.1-23. Иллюстрация к модулю управления Сугено-Такаги

$$\mu_{A_1^1}(4) = 0,25;$$

$$\mu_{A_2^1}(4) = 0,8;$$

$$\mu_{A_1^2}(5) = 1;$$

$$\mu_{A_2^2}(5) = 0,5.$$

Затем в соответствии с правилом Мамдани рассчитываем нечеткую импликацию

$$\alpha_1 = \min(0,25 \ 1) = 0,25;$$

$$\alpha_2 = \min(0,8 \ 0,5) = 0,5.$$

Кроме того рассчитываем «индивидуальные» выходы

$$\bar{z}_1 = f_1(4, 5) = 4 + 2 \cdot 4 - 2 \cdot 5 = 2 ;$$

$$\bar{z}_2 = f_2(4, 5) = 2 + 4 \cdot 4 - 2 \cdot 5 = 8 .$$

Рассчитываем значение z по формуле метода центраида

$$\bar{z} = \frac{\sum_{k=1}^N \alpha_k \bar{z}_k}{\sum_k \alpha_k} = \frac{0,25 \cdot 2 + 0,5 \cdot 8}{0,25 + 0,5} = 6.$$

Пример 1-22. Рассмотрим контур управления напряжением бесщеточного синхронного генератора (БЩСГ) с применением нечеткого регулятора, реализующего, например, алгоритм Сугено-Такаги, показанный на рис.1-24 [8].

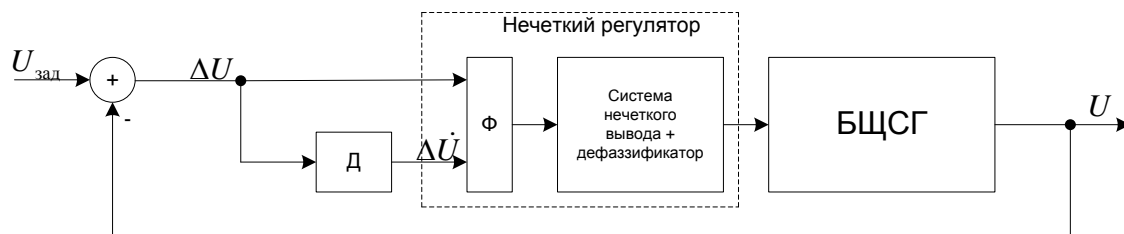


Рис. 1-24. Структурная схема САУ с нечетким регулятором напряжения БЩСГ

Линейные терм-множества фаззификатора регулятора приведены на рис.1-25.

Пусть текущее значение отклонения напряжения $\Delta U_i = -0,17$ активизирует терм Н степенью принадлежности 0,66 и терм ОМ – степенью принадлежности 0,34.

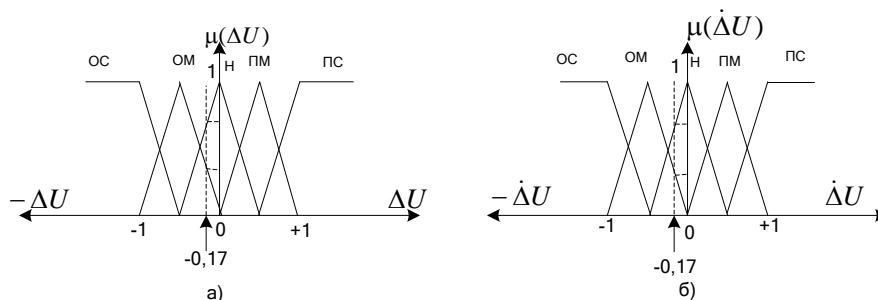


Рис.1-25 Линейное терм-множество фаззификатора

а) по отклонению напряжения; б) по производной отклонения напряжения

$$\mu(\Delta U_i)_H = \delta(\Delta U_i) \cap A_1^3 = 0,66;$$

$$\mu(\Delta U_i)_{OM} = \delta(\Delta U_i) \cap A_1^2 = 0,34,$$

где $\delta(\Delta U_i)$ – дельта-функция, где ΔU_i – текущее значение отклонения напряжения: $\Delta U_i = -0,17$; $\mu(\Delta U_i)_H$ – степень принадлежности $\Delta U_i \in A_1^3$ (терм Н); $\mu(\Delta U_i)_{OM}$ – степень принадлежности $\Delta U_i \in A_1^2$ (терм ОМ).

Пусть текущее значение производной отклонения напряжения $\dot{\Delta U}_i = -0,17$ активизирует терм Н степенью принадлежности 0,66 и терм ОМ – степенью принадлежности 0,34 или как показано на рис.1-25.

$$\mu(\dot{\Delta U}_i)_H = \delta(\dot{\Delta U}_i) \cap A_2^3 = 0,66;$$

$$\mu(\dot{\Delta U}_i)_{OM} = \delta(\dot{\Delta U}_i) \cap A_2^2 = 0,34,$$

где $\delta(\dot{\Delta U}_i)$ – дельта-функция, где $\dot{\Delta U}_i$ – текущее значение производной отклонения напряжения: $\dot{\Delta U}_i = -0,17$; $\mu(\dot{\Delta U}_i)_H$ – степень принадлежности $\dot{\Delta U}_i \in A_2^3$ (терм Н); $\mu(\dot{\Delta U}_i)_{OM}$ – степень принадлежности $\dot{\Delta U}_i \in A_2^2$ (терм ОМ).

Находим степени принадлежности после операции минимум, где нечеткая импликация T -типа (по Мамдани) определяется по формуле:

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cap \mu_B(y) = \min[\mu_A(x), \mu_B(y)]$$

или

$$\alpha_1 = \min [\mu(\Delta U_i)_H, \mu(\dot{\Delta U}_i)_H] = \min [0,66, 0,66] = 0,66;$$

$$\alpha_2 = \min [\mu(\Delta U_i)_H, \mu(\dot{\Delta U}_i)_{OM}] = \min [0,66, 0,34] = 0,34;$$

$$\alpha_3 = \min [\mu(\Delta U_i)_{OM}, \mu(\dot{\Delta U}_i)_H] = \min [0,34, 0,66] = 0,34;$$

$$\alpha_4 = \min [\mu(\Delta U_i)_{OM}, \mu(\dot{\Delta U}_i)_{OM}] = \min [0,34, 0,34] = 0,34.$$

и определяем «индивидуальные выходы»:

$$z_1^* = \mu(\Delta U_i)_H \Delta U_i + \mu(\dot{\Delta U}_i)_H \dot{\Delta U}_i = 0,66(-0,17) + 0,66(-0,17) = -0,22;$$

$$z_2^* = \mu(\Delta U_i)_H \Delta U_i + \mu(\dot{\Delta U}_i)_{OM} \dot{\Delta U}_i = 0,66(-0,17) + 0,34(-0,17) = -0,17;$$

$$z_3^* = \mu(\Delta U_i)_{OM} \Delta U_i + \mu(\dot{\Delta U}_i)_H \dot{\Delta U}_i = 0,34(-0,17) + 0,66(-0,17) = -0,17;$$

$$z_4^* = \mu(\Delta U_i)_{\text{OM}} \Delta U_i + \mu(\dot{\Delta} U_i)_{\text{OM}} \dot{\Delta} U_i = 0,34(-0,17) + 0,34(-0,17) = -0,12.$$

По методу центраида определяем выход.

$$z_{0_{\text{напр}}} = \frac{\sum_{i=1}^4 \alpha_i z_i^*}{\sum_{i=1}^4 \alpha_i} = \frac{0,66(-0,22) + 2 \cdot 0,34(-0,17) + 0,34(-0,12)}{2(0,66 + 0,34)} = -0,38.$$

Иерархические системы нечеткого логического вывода

Для моделирования многомерных зависимостей «входы – выход» целесообразно использовать иерархические системы нечеткого вывода [6]. В таких системах выход одной базы знаний подается на вход другой, более высокого уровня иерархии. В таких системах отсутствуют обратные связи. Применение иерархических нечетких баз знаний позволяет преодолеть «проклятие размерности». При большом количестве входов эксперту трудно описать нечеткими правилами причинно-следственные связи. Это обусловлено тем, что в оперативной памяти человек может одновременно хранить не более 7 ± 2 понятий-признаков. При большом количестве информации ее необходимо иерархически классифицировать, так как человек перерабатывает информацию по иерархическому принципу.

Вторым преимуществом иерархических баз знаний – компактность. На рис. 1-26 показана иерархическая система нечеткого вывода.

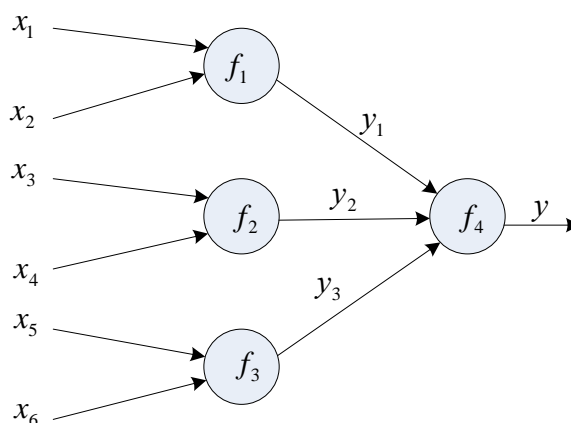


Рис.1-26. Иерархическая система нечеткого вывода

Небольшим количеством нечетких правил в иерархических базах знаний можно адекватно описать многомерные зависимости «вход-выход». Пусть для пяти лингвистических переменных используется по шесть терм. Тогда максимальное количество правил для задания зависимости $y = f(x_1, x_2, x_3, x_4, x_5, x_6)$ с помощью одной базы знаний будет равно $5^6 = 15625$. Как показано для иерархической базы знаний количество правил будет $5^2 + 5^2 + 5^2 + 5^3 = 200$.

1.10. Деффазификация. Способы реализации

Деффазификация – операция перевода нечеткой информации в четкую информацию [9]. Известны методы деффазификации; центр тяжести (*centroid*), центр медианы (*bisector*), наибольшего из максимумов (*lom*), наименьшего из максимумов (*som*) и центр максимумов (*tom*). На практике чаще применяют метод центроида и центр максимумов. Рассмотрим модифицированный метод центроида с целью сокращения объема вычислений и повышения быстродействия.

Деффазификация модифицированным методом центроида.

Необходимо определить минимальную координату абсцисс, соответствующей максимальной высоте (ординате) нелинейного объединенного усеченного множества после нечеткой импликации и нечеткой композиции.

Известны разные формулы расчета координат ЦТ фигуры по методу центроида. Одна из них (1.5) [10].

$$x_{\text{ЦТ}} = \frac{\int x \cdot \mu_{\Sigma}(x) dx}{\int \mu_{\Sigma}(x) dx}. \quad (1.5)$$

Формула (1.5) удобна в случае, когда функции принадлежности деффазификатора есть синглетоны.

Известна формула (1.6) вычисления абсциссы ЦТ линейных объединенных усеченных множеств (фигуры), с фиксацией координат ее характерных точек [11].

$$x_{\text{ЦТ}}^{\Sigma} = \frac{\sum_{i=1}^{n-1} 0,5[y_i(x_{i+1}^2 - x_i^2) - 0,5x_i(x_{i+1} + x_i)(y_{i+1} - y_i) + \frac{1}{3} \cdot \frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)}(x_{i+1}^3 - x_i^3)]}{\sum_{i=1}^{n-1} 0,5(x_{i+1} - x_i)(y_{i+1} + y_i)}, \quad (1.6)$$

где x_i, y_i – координаты характерных точек элементарных фигур, определяющих границы i -результатирующего нечеткого множества (фигуры).

Расчет координат ЦТ фигуры по формуле (1.6) требует знания координат характерных точек элементарных фигур и имеет фиксированную точность, которая зависит от формы фигуры.

Далее приводятся формулы (1.7) и (1.8) определения координат ЦТ фигуры, которые предполагают равномерное разбиение фигуры по оси абсцисс с построением i -прямоугольников разной высоты с расчетом центра тяжести каждого прямоугольника [9].

$$x_{\text{ЦТ}}^{\Sigma} = \frac{\sum S_i x_{\text{ЦТ}i}}{\sum S_i}, \quad (1.7)$$

$$y_{\text{ЦТ}}^{\Sigma} = \frac{0,5 \cdot \sum f(y_i) S_i}{\sum S_i}, \quad (1.8)$$

где $S_i = f(x_i) \cdot \Delta x_i$ – площадь i -прямоугольника; $f(x_i)$ – высота i -прямоугольника; Δx_i – ширина i -прямоугольника; $x_{\text{ЦТ}i} = \frac{x_{i+1} - x_i}{2}$ – значение абсциссы ЦТ i -прямоугольника.

Недостатком применения формул (1.7) и (1.8) является большой объем вычислений и снижение быстродействия формирования управляющего воздействия на объект.

С целью повышения быстродействия расчета координат ЦТ и снятия ограничений на характер функции $y = f(x)$, рассмотрим приближенный алгоритм на основе неравномерного разбиения по оси абсцисс фигуры

объединенного усеченного множества с построением i -прямоугольников равной площади [20].

Пусть объединенное усеченное множество описывается функцией $y = f(x)$ и сложная фигура расположена на интервале $[a, b]$ по оси абсцисс.

Предлагается следующий алгоритм вычисления координат ЦТ фигуры:

– площадь $S = 1$;

– разбиваем сложную фигуру на n простых фигур с неравномерным разбиением по оси абсцисс, причем площади этих фигур должны быть одинаковыми и равными $S \cdot n^{-1}$. Число разбиений n определяется из соотношения $n = (2\Delta F)^{-1}$, где ΔF – абсолютная погрешность воспроизведения функции $y = f(x)$;

– задаемся приращением ΔC изменения координаты абсциссы, которое должно быть хотя бы на порядок меньше, чем $(b-a) \cdot n^{-1}$;

– определяем координаты точек абсцисс в интервале разбиения $[x_k, x_{k+1}]$, где $x_0 = a$; $x_n = b$; $k = 0, 1, 2, \dots, n-1$. Для определения точек разбиения x_i

подсчитываются значение площади $S_i = \sum_{j=1}^k \Delta C f(x_i + j\Delta C)$ и сравниваются со

значением $S \cdot n^{-1}$. Увеличиваем значение k до выполнения условия $(S_i - S \cdot n^{-1}) \leq \varepsilon_{\text{зп}}$, где $\varepsilon_{\text{зп}}$ – заданная точность. Граница i -интервала

определяется по формуле $x_{i+1} = x_i + k\Delta C$. Указанная процедура выполняется на всем отрезке $[a, b]$ и определяет координаты $x_0, x_1, x_2, \dots, x_i, \dots, x_n$. В пределах

каждого интервала $[x_i, x_{i+1}]$ функция $f(x)$ аппроксимируется постоянными значениями y_i ;

– определяем абсциссу и ординату ЦТ объединенного усеченного множества по формулам (1.9) и (1.10), соответственно:

$$x_{\text{ЦТ}}^{\Sigma} = \frac{1}{n} \sum_{i=0}^{n-1} \left(x_i + \frac{x_{i+1} - x_i}{2} \right); \quad (1.9)$$

$$y_{\text{ЦТ}}^{\Sigma} = \frac{0,5}{n} \sum_{i=0}^{n-1} \frac{S_i}{x_{i+1} - x_i}, \quad (1.10)$$

где n – число разбиений; S_i – площади текущих разбиений; x_i, x_{i-1} – координаты абсцисс текущих разбиений.

Приведенный алгоритм вычисления координат ЦТ был проверен на фигуре объединенного усеченного множества, показанного на рис. 1-27.

Проведено сравнение результатов с точными значениями. Исследование было проведено с помощью разработанного пакета «Центр тяжести».

Результаты расчета

Фигура (а).

$$n = 4; \quad x_{ц.т.} = 3,405;$$

$$y_{ц.т.} = 0,330;$$

$$n = 8; \quad x_{ц.т.} = 3,419;$$

$$y_{ц.т.} = 0,330;$$

$$n = 10; \quad x_{ц.т.} = 3,371;$$

$$y_{ц.т.} = 0,328;$$

$$\Delta C = 0,02;$$

$$\delta = 2,28\%;$$

$$\delta = 1,98\%;$$

$$\delta = 1,88\%;$$

$$\delta = 1,98\%;$$

$$\delta = 3,25\%;$$

$$\delta = 2,57\%;$$

$$n = 4; \quad x_{ц.т.} = 3,438;$$

$$y_{ц.т.} = 0,305;$$

$$n = 8; \quad x_{ц.т.} = 3,467;$$

$$y_{ц.т.} = 0,328;$$

$$n = 10; \quad x_{ц.т.} = 3,435;$$

$$y_{ц.т.} = 0,308;$$

$$\Delta C = 0,01;$$

$$\delta = 1,34\%;$$

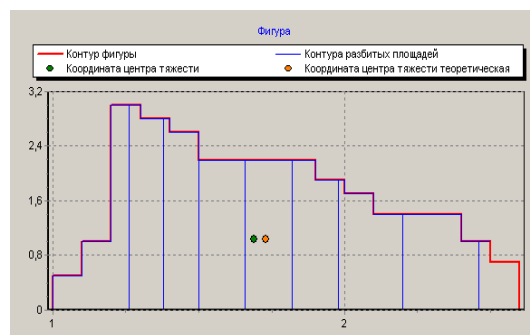
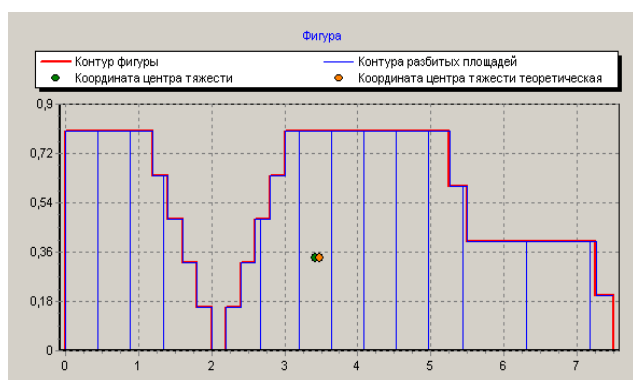
$$\delta = 9,40\%;$$

$$\delta = 5,0\%;$$

$$\delta = 2,71\%;$$

$$\delta = 1,43\%;$$

$$\delta = 8,51\%;$$



а)

б)

Рис.1-27. Примеры фигур: а), б)

Фигура (б).

 $\Delta C = 0,02;$

$n = 4;$	$x_{ц.т} = 1,688;$	$\delta = 2,51%;$
	$y_{ц.т} = 0,988;$	$\delta = 4,11%;$
$n = 8;$	$x_{ц.т} = 1,691;$	$\delta = 2,29%;$
	$y_{ц.т} = 1,044;$	$\delta = 1,36%;$
$n = 10;$	$x_{ц.т} = 1,670;$	$\delta = 3,52%;$
	$y_{ц.т} = 1,065;$	$\delta = 3,42%;$

 $\Delta C = 0,01;$

$n = 4;$	$x_{ц.т} = 1,713;$	$\delta = 1,06%;$
	$y_{ц.т} = 0,925;$	$\delta = 10,18%;$
$n = 8;$	$x_{ц.т} = 1,705;$	$\delta = 1,49%;$
	$y_{ц.т} = 1,013;$	$\delta = 1,68%;$
$n = 10;$	$x_{ц.т} = 1,690;$	$\delta = 2,39%;$
	$y_{ц.т} = 1,050.$	$\delta = 1,96%.$

Анализ расчетных формул и экспериментальных данных показал:

- чем меньше приращение абсциссы ΔC , тем выше точность определения координат ЦТ фигуры объединенного усеченного множества;
- быстроедействие алгоритма для вычисления координат ЦТ фигуры объединенного усеченного множества по формулам (1.9) и (1.10) на два порядка выше, чем при расчетах по формулам (1.7) и (1.8);
- формула расчета координат ЦТ объединенного линейного усеченного множества (1-3) предполагает линейные функции принадлежности дефаззификатора с фиксацией координат характерных точек элементарных фигур;
- формулы расчета координат ЦТ (1.9) и (1.10) объединенного усеченного множества предполагают как линейные, так и нелинейные функции принадлежности дефаззификатора, являются универсальными и могут широко применяться при проектировании нечеткого регулятора.

1.11. Алгоритмы нечеткого управления в системе реального времени

Модуль нечеткого управления состоит из следующих компонентов: блока фаззификации, базы правил, блока выработки решения и блока дефаззификации.

Блок фаззификации.

Система управления с нечеткой логикой оперирует нечеткими множествами. В задачах управления чаще всего применяется операция фаззификации типа синглетон. Блок фаззификации содержит от одного до нескольких терм-множеств. Поэтому конкретному значению $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^T \in X$ входного сигнала блока нечеткого управления после операции фаззификация будет сопоставлено нечеткое множество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$. Переменным сигналом блока фаззификации является унимодальная функция принадлежности (синглетон), которая *синхронно* с входным сигналом перемещается в нормированном интервале изменения входной величины, пересекая терм-множества. Фаззификация на синглетонной базе (операция перевода четкой информации в нечеткую информацию) в случае нечеткого управления формирует нечеткое множество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$ для дальнейшей обработки операцией нечеткая импликация в блоке выработки решения. Число терм отдельного терм-множества задается от 3 до 11 в зависимости от скорости изменения сигнала. Увеличение числа терм возможно включением квантификаторов: НЕ, ОЧЕНЬ, БОЛЕЕ-МЕНЕЕ. Характер терм задаются синтаксическими правилами, а смысл – семантическими правилами.

База правил.

База правил, иногда называемой лингвистической моделью, представляет собой множество нечетких правил $R^{(k)}$, $k=1, \dots, N$, вида (для случая нескольких выходов)

$R^k : IF(x_1 \text{ это } A_1^k \text{ AND } \dots \text{ AND } x_n \text{ это } A_n^k) THEN(y_1 \text{ это } B_1^k \text{ AND } y_2 \text{ это } B_2^k \text{ AND } \dots \text{ AND } y_m \text{ это } B_m^k)$

где $A_i^k \subseteq X_i \subset R$, $i=1, \dots, n$; $x_1, x_2 \dots x_n$ – входные переменные лингвистической модели;

$B_j^k \subseteq Y_j \subset R$, $j=1, \dots, m$; y_1, \dots, y_m – выходные переменные лингвистической модели.

Примем, что выходные переменные y_1, \dots, y_m взаимно независимы, а также входные и выходные переменные могут принимать как лингвистические переменные: «малый», «средний» и «большой», так и числовые значения.

При проектировании модулей нечеткого управления следует оценивать достаточность количества нечетких правил, их непротиворечивость и наличие корреляции между отдельными правилами.

Блок нечеткого вывода.

Допустим, что на вход блока выработки решения подано нечеткое множество $A' \subseteq X = X_1 \times X_2 \times \dots \times X_n$. На выходе этого блока также появится соответствующее одно нечеткое множество B' . Случай, когда выходов несколько из рассмотрения опущен.

Пример 1-24. Рассмотрим вариант математического описания нечеткого регулятора, где нечеткая импликация выполнена с помощью расширения Заде, а фаззификация выполнена на синглетонной базе

Рассмотрим модуль нечеткого управления с базой правил:

$$R^1 : IF(x_1 \text{ это } A_1^1 \text{ AND } x_2 \text{ это } A_2^1) THEN(y \text{ это } B^1);$$

$$R^2 : IF(x_1 \text{ это } A_1^2 \text{ AND } x_2 \text{ это } A_2^2) THEN(y \text{ это } B^2).$$

На его вход подан сигнал $\bar{X} = (\bar{x}_1, \bar{x}_2)^T$. После выполнения фаззификации на синглетонной базе на входе блока выработки решения получаем нечеткие подмножества A'_1 и A'_2 с функциями принадлежности

$$\mu_{A'_1}(x_1) = \delta(x_1 - \bar{x}_1);$$

$$\mu_{A_2'}(x_2) = \delta(x_2 - \bar{x}_2).$$

Обозначим выходной сигнал модуля нечеткого управления символом \bar{y} .

Тогда функция принадлежности нечеткого множества имеет вид

$$\mu_{\bar{B}^k}(y) = \sup_{x_1, x_2} [\min(\mu_{A_1' \times A_2'}(x_1, x_2), \mu_{R^k}(x_1, x_2, y))].$$

В качестве T -нормы применим правило (расширение) Заде. Кроме того допустим, что

$$\mu_{A_1' \times A_2'}(x_1, x_2) = \min[\mu_{A_1'}(x_1), \mu_{A_2'}(x_2)] = \min[\delta(x_1 - \bar{x}_1), \delta(x_2 - \bar{x}_2)].$$

В этом случае

$$\mu_{\bar{B}^k}(y) = \sup_{x_1, x_2} [\min(\delta(x_1 - \bar{x}_1), \delta(x_2 - \bar{x}_2), \mu_{R^k}(x_1, x_2, y))] = \mu_{R^k}(\bar{x}_1, \bar{x}_2, y).$$

В качестве выбранного варианта нечеткой импликации

$$\mu_{R^k}(\bar{x}_1, \bar{x}_2, y) = \mu_{A_1^k \times A_2^k \rightarrow B^k}(\bar{x}_1, \bar{x}_2, y).$$

использована операция минимум

$$\mu_{A_1^k \times A_2^k \rightarrow B^k}(\bar{x}_1, \bar{x}_2, y) = \min[\mu_{A_1' \times A_2'}(\bar{x}_1, \bar{x}_2), \mu_{B^k}(y)],$$

где

$$\mu_{A_1' \times A_2'}(\bar{x}_1, \bar{x}_2) = \min[\mu_{A_1'}(\bar{x}_1), \mu_{A_2'}(\bar{x}_2)].$$

В результате

$$\mu_{\bar{B}^k}(y) = \min\{\min[\mu_{A_1'}(\bar{x}_1), \mu_{A_2'}(\bar{x}_2)], \mu_{B^k}(y)\} = \min[\mu_{A_1'}(\bar{x}_1), \mu_{A_2'}(\bar{x}_2), \mu_{B^k}(y)].$$

Окончательное выражение математического описания нечеткого регулятора имеет вид

$$\mu_{B'}(y) = \max_{K=1,2} \{\min[\mu_{A_1^k}(\bar{x}_1), \mu_{A_2^k}(\bar{x}_2), \mu_{B^k}(y)]\}.$$

Пример 1-25. Рассмотрим вариант математического описания нечеткого регулятора, где нечеткая импликация выполнена по Мамдани, а фаззификация выполнена на несинглетонной базе

Пусть модуль нечеткого управления имеет базу правил:

$$R^1 : IF(x_1 \text{ это } A_1^1 \text{ AND } x_2 \text{ это } A_2^1) THEN (y \text{ это } B^1);$$

$$R^2 : IF(x_1 \text{ это } A_1^2 \text{ AND } x_2 \text{ это } A_2^2) THEN (y \text{ это } B^2).$$

На его вход поданы числовые входные сигналы \bar{x}_1 и \bar{x}_2 . После фаззификации на входе блока выработки решения появляются нечеткие подмножества A_1' и A_2' с функциями принадлежности $\mu_{A_1'}(x_1)$ и $\mu_{A_2'}(x_2)$. В данном примере фаззификация выполняется на несинглетонной базе.

Обозначим выходной сигнал модуля нечеткого управления символом \bar{y} .

Тогда функция принадлежности нечеткого подмножества имеет вид

$$\mu_{\bar{B}^k}(y) = \sup_{x_1, x_2} [\min(\mu_{A_1' \times A_2'}(x_1, x_2), \mu_{R^k}(x_1, x_2, y))].$$

В качестве T - нормы примем правило Мамдани

$$\mu_{A_1' \times A_2'}(x_1, x_2) = \min[\mu_{A_1'}(x_1), \mu_{A_2'}(x_2)].$$

В этом случае

$$\mu_{\bar{B}^k}(y) = \sup_{x_1, x_2} \{\min[\min(\mu_{A_1'}(x_1), \mu_{A_2'}(x_2)), \mu_{R^k}(x_1, x_2, y)]\} = \mu_{R^k}(\bar{x}_1, \bar{x}_2, y).$$

В качестве нечеткой импликации

$$\mu_{R^k}(\bar{x}_1, \bar{x}_2, y) = \mu_{A_1^k \times A_2^k \rightarrow B^k}(\bar{x}_1, \bar{x}_2, y).$$

используем операцию минимум

$$\mu_{A_1^k \times A_2^k \rightarrow B^k}(\bar{x}_1, \bar{x}_2, y) = \min[\mu_{A_1' \times A_2'}(\bar{x}_1, \bar{x}_2), \mu_{B^k}(y)].$$

Кроме того

$$\mu_{A_1' \times A_2'}(\bar{x}_1, \bar{x}_2) = \min[\mu_{A_1^k}(\bar{x}_1), \mu_{A_2^k}(\bar{x}_2)].$$

В результате

$$\mu_{\bar{B}^k}(y) = \min\{\min[\mu_{A_1^k}(\bar{x}_1), \mu_{A_2^k}(\bar{x}_2)], \mu_{B^k}(y)\} = \min[\mu_{A_1^k}(\bar{x}_1), \mu_{A_2^k}(\bar{x}_2), \mu_{B^k}(y)].$$

Окончательное выражение математического описания нечеткого регулятора, используя алгоритм Мамдани, имеет вид

$$\mu_{B'}(y) = \max_{K=1,2} \left\{ \min [\mu_{A_1^k}(\bar{x}_1), \mu_{A_2^k}(\bar{x}_2), \mu_{B^k}(y)] \right\}.$$

Иллюстрация графической интерпретация нечеткого вывода по Мамдани, где фаззификация выполнена на несинглетонной базе (см. рис.8) [5].

Рассмотрим алгоритм реализации нечеткого управления в системах реального времени.

Синтез нечеткого регулятора, в общих чертах, заключается в выборе функций принадлежности терм-множеств лингвистических переменных, алгоритмы нечеткого вывода, оптимизации основных параметров регулятора (диапазонов изменения лингвистических переменных, формы и параметров функций принадлежности) путем минимизации выбранного критерия качества в замкнутой системе автоматического регулирования. Искусство проектировщика состоит в том, чтобы выбрать для каждой конкретной фаззи-системе автоматического управления не только функции принадлежности (сколько и каких) в нечетком регуляторе, но и тип регулятора, т.е. решить вопрос, какие составляющие ошибки, кроме самой ошибки, подавать на вход регулятора, а также требования по быстродействию в системах реального времени.

Рассмотрим поэтапно алгоритм проектирования нечеткого регулятора, который включает этапы фаззификации, нечеткого вывода и дефаззификации для управления регулируемым объектом.

1. Формирование проекций входных векторов после фаззификации нечеткого регулятора.

Для формирования нечеткого множества на синглетонной базе используется унимодальная функция принадлежности (синглетон), который жестко связан с входной переменной. Для случая двух лингвистических переменных зададимся терм-множеством лингвистических переменных (рис.1-29): отклонение частоты вращения свободной турбины и ее производной.

В процессе фаззификации терм-множества отклонения частоты вращения свободной турбины на выходе формируется нечеткое множество из степеней принадлежности вида:

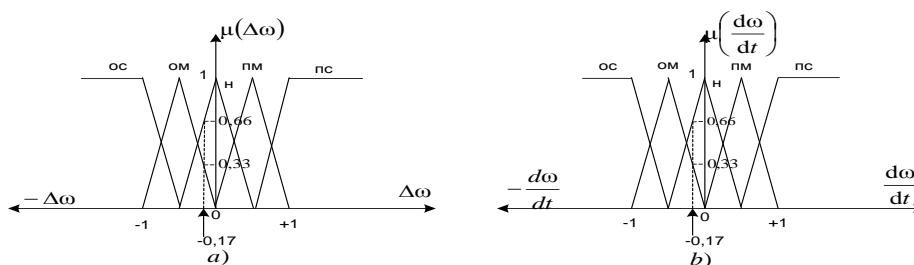


Рис.1-28. Терм-множество фаззификатора нечеткого регулятора, например частоты: а – терм-множество по отклонению частоты вращения турбины; б – терм-множество по производной частоты вращения турбины

В процессе фаззификации терм-множества отклонения частоты вращения свободной турбины на выходе формируется нечеткое множество из степеней принадлежности вида:

$$\mu_{A_1}(x_1) = \delta(x_1^1 - \bar{x}_1)$$

$$\mu_{A_2}(x_1) = \delta(x_1^2 - \bar{x}_1)$$

$$\mu_{A_3}(x_1) = \delta(x_1^3 - \bar{x}_1)$$

$$\mu_{A_4}(x_1) = \delta(x_1^4 - \bar{x}_1)$$

$$\mu_{A_5}(x_1) = \delta(x_1^5 - \bar{x}_1).$$

Данное нечеткое множество можно трактовать как проекции входного вектора x_1 (например, отклонения частоты вращения свободной турбины).

В процессе фаззификации терм-множества производной входного вектора (например, производная отклонения частоты вращения свободной турбины) на выходе формируется нечеткое множество степеней принадлежности вида:

$$\mu_{A_2}(x_2) = \delta(x_2^1 - \bar{x}_2)$$

$$\mu_{A_2}(x_2) = \delta(x_2^2 - \bar{x}_2)$$

$$\mu_{A_3}(x_2) = \delta(x_2^3 - \bar{x}_2)$$

$$\mu_{A_2^4}(x_2) = \delta(x_2^4 - \bar{x}_2)$$

$$\mu_{A_2^5}(x_2) = \delta(x_2^5 - \bar{x}_2).$$

Данное нечеткое множество можно трактовать как производные проекций входного вектора x_2 .

2. Нечеткая импликация

После определения нечетких множеств по входным переменным необходимо формировать матрицу решений, используя предиктивные правила, составленные экспертом [6]. В виду сложности определения предиктивных правил предлагается следующая методика формирования матрицы решений:

– необходимо задать отдельно терм-множества входных лингвистических переменных

A_1^1	A_1^2	A_1^3	A_1^4	A_1^5
---------	---------	---------	---------	---------

и

A_2^1	A_2^2	A_2^3	A_2^4	A_2^5
---------	---------	---------	---------	---------

– необходимо задать терм-множество выхода, например $(B^1, B^2, B^3, B^4, B^5)$;

– расположить терм B^3 лингвистической переменной выхода по диагонали матрицы;

– расположить термы B^1, B^2, B^4, B^5 лингвистической переменной выхода матрицы решений 5×5 , как показано на рис. 1-29.

3. Нечеткая импликация, например по Ларсену.

Определить функции принадлежности как произведение терм лингвистической переменной выхода на соответствующие α -срезы:

$$\mu_{\bar{B}^1}(y) = B^5 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^1}(\bar{x}_2)]; \quad \mu_{\bar{B}^2}(y) = B^4 * \min[\mu_{A_1^2}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)];$$

$$\mu_{\bar{B}^3}(y) = B^3 * \min[\mu_{A_1^3}(\bar{x}_1), \mu_{A_2^3}(\bar{x}_2)]; \quad \mu_{\bar{B}^4}(y) = B^2 * \min[\mu_{A_1^4}(\bar{x}_1), \mu_{A_2^4}(\bar{x}_2)];$$

$$\mu_{\bar{B}^5}(y) = B^4 * \min[\mu_{A_1^5}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)]; \quad \mu_{\bar{B}^6}(y) = B^5 * \min[\mu_{A_1^2}(\bar{x}_1), \mu_{A_2^1}(\bar{x}_2)];$$

$$\mu_{\bar{B}^7}(y) = B^4 * \min[\mu_{A_1^3}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)]; \quad \mu_{\bar{B}^8}(y) = B^4 * \min[\mu_{A_1^4}(\bar{x}_1), \mu_{A_2^3}(\bar{x}_2)];$$

Функции принадлежности		x_2				
		A_2^1	A_2^2	A_2^3	A_2^4	A_2^5
x_1	A_1^1	B^5	B^5	B^4	B^4	B^3
	A_1^2	B^5	B^4	B^4	B^3	B^2
	A_1^3	B^4	B^4	B^3	B^2	B^2
	A_1^4	B^4	B^3	B^2	B^2	B^1
	A_1^5	B^3	B^2	B^2	B^1	B^1

Рис. 1-29. Матрица решений 5×5 :

$A_1^1, A_1^2, A_1^3, A_1^4, A_1^5$ - проекции вектора x_1 ; $A_2^1, A_2^2, A_2^3, A_2^4, A_2^5$ - проекции вектора x_2 ;

B^1, B^2, B^3, B^4, B^5 - проекции вектора выхода

$$\mu_{\bar{B}^9}(y) = B^3 * \min[\mu_{A_1^5}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)]; \quad \mu_{\bar{B}^{10}}(y) = B^5 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{11}}(y) = B^4 * \min[\mu_{A_1^3}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)]; \quad \mu_{\bar{B}^{12}}(y) = B^3 * \min[\mu_{A_1^4}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{13}}(y) = B^2 * \min[\mu_{A_1^5}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)]; \quad \mu_{\bar{B}^{14}}(y) = B^4 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^3}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{15}}(y) = B^4 * \min[\mu_{A_1^2}(\bar{x}_1), \mu_{A_2^2}(\bar{x}_2)]; \quad \mu_{\bar{B}^{16}}(y) = B^2 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^3}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{17}}(y) = B^2 * \min[\mu_{A_1^5}(\bar{x}_1), \mu_{A_2^3}(\bar{x}_2)]; \quad \mu_{\bar{B}^{18}}(y) = B^4 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^4}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{19}}(y) = B^3 * \min[\mu_{A_1^2}(\bar{x}_1), \mu_{A_2^4}(\bar{x}_2)]; \quad \mu_{\bar{B}^{20}}(y) = B^3 * \min[\mu_{A_1^3}(\bar{x}_1), \mu_{A_2^4}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{21}}(y) = B^1 * \min[\mu_{A_1^5}(\bar{x}_1), \mu_{A_2^4}(\bar{x}_2)]; \quad \mu_{\bar{B}^{22}}(y) = B^3 * \min[\mu_{A_1^1}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{23}}(y) = B^2 * \min[\mu_{A_1^2}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)]; \quad \mu_{\bar{B}^{24}}(y) = B^2 * \min[\mu_{A_1^3}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)];$$

$$\mu_{\bar{B}^{25}}(y) = B^1 * \min[\mu_{A_1^4}(\bar{x}_1), \mu_{A_2^5}(\bar{x}_2)].$$

4. Нечеткая композиция.

Определить площадь результирующей фигуры методом поточечного суммирования.

$$Y_{\bar{B}} = \sum_{i=1}^{25} \mu_{\bar{B}^i}(y).$$

5. Дефаззификация.

Определить координаты центра тяжести (абсциссу и координату) результирующей фигуры модифицированным методом центраида в случае, когда терм – множество лингвистической переменной выхода содержит сложные функции (Мамдани, Ларсен) [9]

Для случая, когда терм-множество лингвистической переменной выхода есть синглетоны, то дефаззификация нечеткого адаптивного регулятора выполнить по методу центраида [5]

$$x_{cl} = \frac{\sum_i \mu_i Q_i}{\sum_i \mu_i},$$

где i – число терм терм-множеств фаззификатора и дефаззификатора; μ_i – степень принадлежности Q_i – соответствующее значения термы дефаззификатора.

Недостатками нечетких систем являются:

- отсутствие стандартной методики проектирования и расчета нечетких систем;
- невозможность математического анализа нечетких систем существующими методами; применение нечеткого подхода по сравнению с вероятностным подходом не приводит к повышению точности вычислений;
- увеличение входных переменных увеличивает сложность вычислений экспоненциально;
- как следствие предыдущего пункта, увеличивается база правил, что приводит к трудному ее восприятию.

Контрольные вопросы

1. Лингвистическая переменная. Способ задания. Примеры.
2. Классификация функций принадлежности. Примеры.
3. Основные операции нечеткой логики.

4. Четкая и расширенная логика «И».
5. Четкая и расширенная логика «ИЛИ».
6. Четкая и расширенная логика «НЕ».
7. Нечеткая импликация.
8. Фаззификация на синглетонной и несинглетонной базе.
9. Система нечеткого вывода по Мамдани.
10. Система нечеткого вывода по Ларсену.
11. Система нечеткого вывода по Тцукамото.
12. Система нечеткого вывода по Заде.
13. Система нечеткого вывода по Сугено-Такаги.
14. Дефаззификация. Модифицированный метода центроида.
15. Нечеткие числа.
16. Декартовое произведение.
17. Достоинства и недостатки нечеткого управления.
18. Фаззификация по Гостеву.
19. Терм-множество. Степень принадлежности.
20. Формы записи нечетких множеств.
21. Композиционное правило Заде.
22. Нечеткие отношения.
23. Операции с нечеткими отношениями.
24. Нечеткая композиция.
25. Иерархическая база знаний.

Таким образом, рассмотрена теория нечетких множеств: основные понятия, основные операции над нечеткими множествами, нечеткие отношения и основные операции над ними, нечеткая импликация, нечеткая композиция, варианты графической интерпретации работыв блока нечеткого вывода, методы дефаззификации. Отмечены достоинства и недостатки нечеткого управления. Основной недостаток нечеткого управления – отсутствие адаптивности к внешним условиям можно устранить, если

использовать другой вычислительный элемент, например, искусственный нейрон в различных модификациях.

2. Основы теории нечетко-нейронного управления недетерминированными объектами в системах реального времени

2.1. Биологический нейрон. Основные положения теории искусственных нейронных сетей

Нейронной сетью называется динамическая система, состоящая из совокупности связанных между собой по типу узлов направленного графа элементарных процессоров, называемыми формальными нейронами, и способная генерировать выходную информацию в ответ на входное воздействие [4]. Актуальность исследований искусственных нейронных сетей (ИНС) подтверждается многообразием их возможных применений. Теория ИНС является алгоритмическим базисом развития нейрокомпьютеров, подобно тому, как булева алгебра более 50 лет являлась алгоритмическим базисом однопроцессорных и многопроцессорных ЭВМ. Современные цифровые вычислительные машины превосходят человека по способности производить числовые и символьные вычисления. Однако человек может без усилий решать сложные задачи восприятия внешних данных (например, узнавать в толпе знакомого только по его промелькнувшему лицу) с такой скоростью и точностью, что самый мощный в мире компьютер по сравнению с ним окажется безнадежным тугодумом. Причина столь значительного различия в их производительности заключена в том, что архитектура биологической нейронной системы совершенно не похожа на архитектуру машины фон Неймана (табл. 2-1), а это существенно влияет на типы функций, которые более эффективно исполняются каждой из моделей. ИНС – это сеть с конечным числом слоев из однотипных элементов – аналогов нейронов с различными типами связей между слоями. При этом число нейронов

в слоях выбирается исходя из необходимости обеспечения заданного качества решения задачи, а число слоев нейронов – как можно меньшее для сокращения времени решения задачи.

Развитие ИНС вдохновлялось и вдохновляется биологией. Однако наши знания о работе мозга ограничены, и разработчикам ИНС приходится выходить за пределы современных биологических знаний в поисках структур, позволяющих выполнять полезные функции. Так часто создаются сети невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга. Нервная система человека, построенная из элементов, называемых нейронами, очень сложна. Около 10^{11} нейронов участвуют в примерно 10^{15} передающих связях. Уникальными способностями нейронов являются прием, обработка и передача электрохимических сигналов по нервным путям, которые образуют коммуникационную систему мозга. Несмотря на огромное количество нейронов, их тела занимают всего несколько процентов общего объема мозга. Почти все остальное пространство занято межнейронными связями.

Число связей каждого нейрона не имеет аналогов в современной технике. Понимание того, что межнейронные связи относятся к основным структурным компонентам мозга, в первую очередь, определяющим его функциональные характеристики, является одним из наиболее существенных выводов, сделанных нейрофизиологами.

На рис. 2-1 приведена структура соединения двух типичных биологических нейронов.

Дендриты идут от тела нервной клетки (**сома**) к другим нейронам, где они принимают сигналы в точках соединения, называемых **синапсами**.

Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, а другие воспрепятствовать его возбуждению. В соответствии с этим различают процессы синаптического возбуждения и торможения нейрона.

Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону, который на конце сильно ветвится, сигнал другим нейронам. Нервный импульс обладает способностью распространяться по нервным волокнам без затухания, вплоть до синаптических окончаний со скоростью распространения около 10 м/с.

Синаптическое торможение снижает возбуждение нейрона и тем самым затрудняет переход возбуждения нейрона в нервный импульс. После генерации импульса наступает так называемый рефрактерный период, во время которого нейрон восстанавливает способность к генерации следующего импульса. Длительность рефрактерного периода менее 1 мс. Поэтому максимальная частота генерации импульсов нейронов около 1 кГц.

Таблица 2-1. Машина фон Неймана по сравнению с биологической НС

Параметры сравнения	Машина фон Неймана	Биологическая нейронная система
	1	2
Процессор	Сложный	Простой
	Высокоскоростной	Низкоскоростной
	Один или несколько	Большое количество
Память	Отделена от процессора	Интегрирована в процессор
	Локализованная	Распределенная
	Адресация не по содержанию	Адресация по содержанию
Надежность	Высокая уязвимость	Живучесть
Специализация	Численные и символьные операции	Проблемы восприятия
Вычисления	Централизованные	Распределенные
	Последовательные	Параллельные
	Хранимые программы	Самообучение
Среда	Строго определенная	Плохо определенная

Функционирование	Строго ограниченная	Без ограничений
------------------	---------------------	-----------------

Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону, который на конце сильно ветвится, сигнал другим нейронам. Нервный импульс обладает способностью распространяться по нервным волокнам без затухания, вплоть до синаптических окончаний со скоростью распространения около 10 м/с.

Синаптическое торможение снижает возбуждение нейрона и тем самым затрудняет переход возбуждения нейрона в нервный импульс. После генерации импульса наступает так называемый рефрактерный период, во время которого нейрон восстанавливает способность к генерации следующего импульса. Длительность рефрактерного периода менее 1 мс. Поэтому максимальная частота генерации импульсов нейронов около 1 кГц.

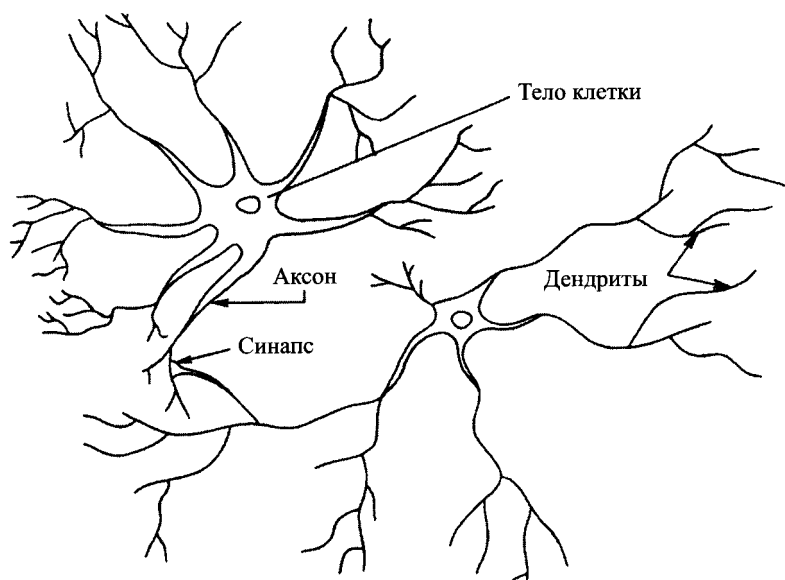


Рис. 2-1. Упрощенная модель биологических нейронов и их соединение

Формальный нейрон

На рис. 2-2 показан формальный нейрон – аналог биологического нейрона.

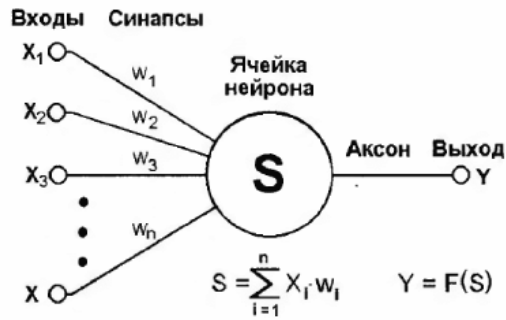


Рис. 2-2. Формальный нейрон

Формальный нейрон обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. Общий вид нейрона, реализующего эти свойства, приведен на рис. 2-2. Здесь множество входных сигналов, обозначенных вектором X . Каждый вес w_i соответствует "силе" одной биологической синаптической связи. Множество весов в совокупности обозначается вектором W . Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически с учетом знака

$$S = \sum_{i=1}^n x_i w_i .$$

Искусственный нейрон

Первое систематическое изучение было предпринято Мак-Каллоком и Питтсом в 1943 г., которые предложили искусственный нейрон, показанный на рис. 2-3 [5].

Пороговая функция имеет вид

$$\text{sign}(g) = \begin{cases} b, & \text{если } g < 0 \\ c, & \text{если } g \geq 0 \end{cases}$$

где b и c – постоянные числа.

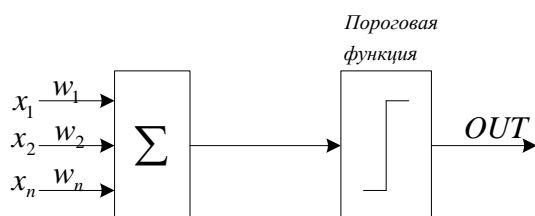


Рис. 2-3. Искусственный нейрон Мак-Каллока и Питтса

На практике чаще всего используют две пары постоянных b и c : первая $(-1,1)$ называемая симметричной пороговой функцией, вторая $(0, 1)$ называется смещенной пороговой функцией.

Математическая модель нейрона с пороговой функцией активации имеет вид:

$$y(t) = \text{sign}(g(t)) = \text{sign} \sum_{i=1}^n (a_i x_i(t) + a_0).$$

Входные сигналы (проекция вектора X) искусственного нейрона арифметически умножаются на синапсы и арифметически суммируются. Если сигнал суммы больше заданного значения порога, то на выходе нейрона 1, если меньше – ноль.

Перцептрон

Сети, составленные из нейронов Мак-Каллока и Питтса, называются перцептроны (восприятие) (1957 г.). Перцептрон Розенблатта (трехслойная сеть, состоящая из однородных искусственных нейронов) показан на рис. 2-4[12].

В Корнельской авиационной лаборатории была разработана модель МАРК-1, которая содержала восьмислойная сеть, на которой проведена серия опытов по распознаванию букв алфавита и геометрических образов.

Недостаток перцептрона – не способность обучаться даже решению ряда простых задач. Однако доказательство теоремы обучения перцептрона показало, что перцептрон способен научиться всему, что он способен представлять. В Корнельской авиационной лаборатории была разработана модель МАРК-1, которая содержала восьмислойная сеть, на которой

проведена серия опытов по распознаванию букв алфавита и геометрических образов.

Недостаток персептрона – не способность обучаться даже решению ряда простых задач. Однако доказательство теоремы обучения персептрона показало, что персептрон способен научиться всему, что он способен представлять.

Дальнейшее развитие нейронных сетей связано с заменой пороговой активационной функции искусственного нейрона другими функциями активации.

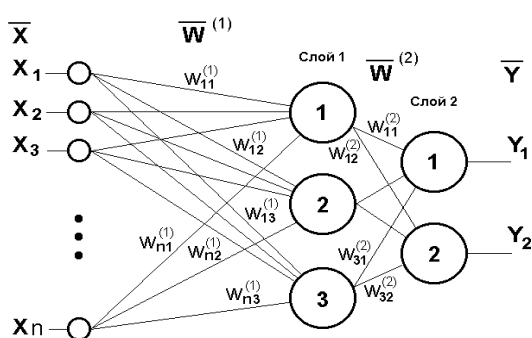


Рис. 2-4. Персептрон

2.2. Функции активации

Сигнал с выхода сумматора искусственного нейрона подается на вход нелинейной функции F . Нелинейная функция F называется активационной и может иметь различный вид.

В качестве активационных функций могут быть использованы:

- степенная функция первого порядка $y = kx$;
- нелинейная динамическая активационная функция, которая включается при выходе из $+ 5\%$ зоны отклонения заданного значения и текущего значения параметра (напряжения), обеспечивая форсировку:

$$K = K_1 \left(1 + |U_{\text{зад}} - U| \right)^{\left(|U_{\text{зад}} - U| K_n \right)}, \text{ при } (U_{\text{зад}} - U) \geq 0;$$

$$K = K_1, \text{ при } (U_{\text{зад}} - U) < 0,$$

где K – динамическая активационная функция;

K_1 – статическая активационная функция;

$K_{п}$ – коэффициент положительного модуля.

- логистические функции различных модификаций.

На рис. 2-5 приведены разновидности функций активаций.

Широко применяемой на практике является сигмоидная функция с насыщением и без смещения, которую предложил Мамдани:

$$f(x) = \frac{1}{1 + e^{-ax}}$$

С уменьшением параметра a сигмоид становится более пологим, в пределе при $a = 0$ вырождаясь в горизонтальную линию на уровне 0,5.

При увеличении a сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке $x = 0$.

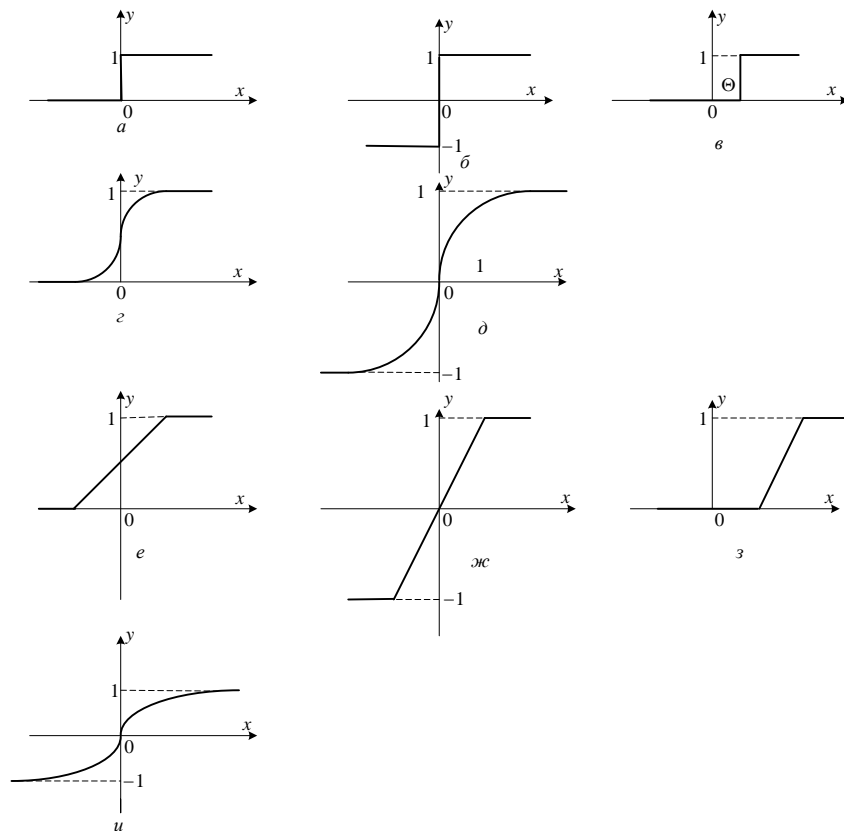


Рис. 2-5. Функции активации: а) пороговая асимметричная; б) пороговая симметричная; в) пороговая асимметричная с зоной нечувствительности; г) сигмоидная; д) гиперболический тангенс; е) линейная асимметричная; ж) линейная симметричная; з) линейная асимметричная с зоной нечувствительности; и) модифицированный гиперболический тангенс.

Из формулы сигмоида очевидно, что выходное значение нейрона лежит в диапазоне $[0,1]$. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной, применение которого будет рассмотрено в дальнейшем,

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

Следует отметить, что *сигмоидная функция дифференцируема* на всей оси абсцисс, что широко используется во многих алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем сильные, и предотвращает насыщение от сильных сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон. Другой широко используемой активационной функцией является гиперболический тангенс. В отличие от логистической функции гиперболический тангенс принимает значения различных знаков, что для ряда сетей оказывается выгодным.

Для смещенной симметричной функции с насыщением выражение имеет вид

$$f[g(t)] = \frac{1 - e^{-g}}{1 + e^{-g}}.$$

Упрощенные сигмоидные функции (гиперболический тангенс)

$$f(r) = th(r) = \frac{e^r - e^{-r}}{e^r + e^{-r}}. \quad f'(r) = (1 + f(r))(1 - f(r))$$

В случае, когда функция активации одна и та же для всех нейронов сети, сеть называют однородной (гомогенной). Если же активационная функция зависит еще от одного или нескольких параметров, значения которые меняются от нейрона к нейрону, то сеть называют неоднородной (гетерогенной).

Говоря о возможной классификации ИНС, важно отметить существование бинарных и аналоговых сетей. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения; логический ноль ("заторможенное" состояние) и логическая единица ("возбужденное" состояние).

В аналоговых сетях выходные значения нейронов способны принимать непрерывные значения. Еще одна классификация делит ИНС на синхронные и асинхронные ИНС. В первом случае в каждый момент времени свое состояние меняет лишь один нейрон. Во втором – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Для программных имитаторов нейронных сетей на цифровых ЭВМ, вопросы, связанные с синхронизацией, решаются компьютером, на котором реализуются ИНС. Рассмотренная простая модель искусственного нейрона существенно упрощает ряд свойств биологического двойника. Например, она не принимает во внимание задержки во времени, которые воздействуют на динамику системы. Входные сигналы сразу же порождают выходной сигнал и, что более важно, она не учитывает воздействие синхронизирующей функции биологического нейрона, которую ряд исследователей считают решающей. Несмотря на эти ограничения, сети, построенные из этих нейронов, обнаруживают свойства, сильно напоминающие биологическую систему. Только время и дальнейшие исследования могут дать ответ на вопрос, являются ли подобные совпадения случайными или это следствие того, что в данной модели верно схвачены основные черты биологического нейрона.

Итак, дано описание основных элементов, из которых составляются нейронные сети. Перейдем теперь к вопросу: как составлять эти сети, как их конструировать? Строгих ограничений здесь нет, как угодно, лишь бы входы получали какие-нибудь сигналы. Возможности безграничны, но обычно используют несколько стандартных архитектур, из которых при некоторых небольших модификациях строят большинство используемых сетей.

2.3. Теоремы Колмогорова–Арнольда и Хехт-Нильсена и их следствия

Построить многомерное отображение $X \rightarrow Y$ – это значит представить его с помощью математических операций над не более, чем с двумя переменными [4]. Ими доказано:

– возможность представления непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных (теорема 1);

– любую непрерывную функцию трех переменных можно представить в виде суммы функций не более двух переменных (теорема 2);

– любые непрерывные функции нескольких переменных можно представить в виде суперпозиций непрерывных функций одного переменного и сложения (теорема 3).

Данная теорема была переложена для нейронных сетей работами Хехт-Нильсена. Теорема доказывает представимость функций многих переменных общего вида с помощью двухслойной нейронной сети с n нейронами входного слоя, $(2n+1)$ нейронами скрытого слоя с заранее известными ограниченными функциями активации (например, сигмоидальными) и m нейронами выходного слоя с неизвестными функциями активации.

Основанием для построения нейронов и ИНС явилась теорема существования, предложенная Мак-Каллоком У.С. и Питтс У. (1943 г.) и две конструктивных теоремы.

Теорема существования утверждает, что любую функцию нервной системы можно описать с помощью конечного числа слоев из искусственных нейронов, то есть реализовать ИНС.

Первая конструктивная теорема говорит, что логические функции *И*, *ИЛИ*, *НЕ* можно вычислить с помощью двухслойной нейронной сети.

Вторая конструктивная теорема говорит, что любую логическую функцию относительно n бинарных элементов можно вычислить с помощью нейронной сети (которая использует в качестве компонентов элементы *И*, *ИЛИ*, *НЕ*).

2.4. Разновидности и классификация нейронных сетей

Нейрон есть элемент, выполняющий взвешенное суммирование значений координат входного вектора с последующим нелинейным / линейным преобразованием суммы. Входные сигналы $x_1 \dots x_n$ через блоки весовых коэффициентов w_n (синапсы) подаются на сумматор вместе со смещающим сигналом x_0 . Результирующий сигнал r подается на вход активационной функции F , единственный выход которого в нейрофизиологии называют аксоном. Степень «интеллекта» одиночного нейрона очень незначительна. Ощутимый эффект в обработке данных может быть достигнут только использованием множества взаимодействующих между собой адаптивных нейронов. Нейроны группируются в нейронные слои и образуют искусственные нейронные сети (ИНС). ИНС индуцированы биологией, так как в некоторой степени соответствуют анатомии мозга, которому присуще [13]:

- массовый параллелизм, т.е. параллельная обработка информации одновременно всеми нейронами, что допускает обработку сигналов в реальном времени;
- способность к обучению и обобщению;
- адаптивность;
- толерантность (нечувствительность) к ошибкам;
- низкое энергопотребление.

Классификация стандартных нейронных сетей

Многослойные ИНС делятся на: сети без обратной связи (ОС), ИНС с ОС и гибридные.

По типам структур нейронов ИНС делятся на: однородные (гомогенные) и гетерогенные (неоднородные) структуры.

По выходному сигналу ИНС делятся на: системы с бинарным и аналоговым выходами.

По времени срабатывания ИНС делятся на: синхронные и асинхронные системы.

В общем случае ИНС рассматривается как направленные графы сигналов, у которых узлами являются искусственные нейроны, а ветви – коэффициенты между узлами. По архитектуре связей ИНС разделяются на два класса:

- сети прямого действия или статические сети;
- сети с обратными связями или динамические сети.

Комплекты нейронов, разделенные на слои, образуют многослойную ИНС.

Многослойная ИНС состоит:

- из входного слоя (входные сигналы) с весовыми коэффициентами, устанавливаемыми с целью масштабирования входных величин;
- скрытых (промежуточных) слоев, определяемых эвристическим путем, исходя их объема области знаний и сложности задачи регулирования;
- из выходного слоя.

Рассмотрим построение стандартных нейронных сетей на базе искусственных нейронов Мак-Каллока и Питтса, каждый из которых выполняет арифметические операции умножения и сложения.

2.4.1. Нейронные сети без обратных связей

Сети прямого распространения (Feedforward)

Сети прямого распространения применяются для распознавания образов, решать задачи предсказания временных рядов, финансовых рисков, адаптивного управления, цифровой обработки и сжатия сигналов и т.д. Многослойные нейронные сети способны с различной точностью аппроксимировать любую из существующих логических функций.

От архитектуры межнейронных связей зависит интеллект, класс решаемых задач, сложность технической реализации, принципы построения алгоритмов обучения и т.д. Один из первых предложенных моделей

структурной организации нейронной сети был персептрон Розенблатта, который представлял собой однослойную сеть, построенную из формальных нейронов. Все связи направлены строго от входных нейронов к выходным нейронам.

2.5. Алгоритмы обучения с учителем

2.5.1. Алгоритм обратного распространения ошибки

Обучение с помощью ОРО требует выполнения следующих действий [4, 12]:

- выбрать очередную обучающую пару обучающего множества и подать на вход сети;
- вычислить выход сети;
- вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары);
- подкорректировать веса сети так, чтобы минимизировать ошибку;
- повторять предыдущие шаги для каждого обучающего вектора до тех пор пока, пока на всем множестве не достигла приемлемого уровня.

Пример 2-1. Рассмотрим вариант НС, содержащий входной (распределительный) слой НС, который отражает вектор степеней принадлежности терм фазсификатора по отклонению, 1 соответствующий терме ПМ, 2 – терме Н, 7 – терме ОМ и по скорости отклонения 8 – соответствующей терме ПМ, 9 – терме Н, 10 – терме ОМ. Внутренний слой НС состоит из нейронов 4 и 5 с обучаемым смещением и выходной слой – из нейрона 6 с обучаемым смещением. Обучение НС выполним методом ОРО. Примем целевое значение выхода НС: $t_j = 0,9$. Условие сходимости $\varepsilon = 0,01$. Начальные значения коэффициентов матриц НС принять в диапазоне от $-0,3$ до $+0,3$, а скорость обучения НС $\eta = 0,25$.

Примем матрицу коэффициентов внутреннего слоя НС

$$w_1 = \begin{pmatrix} 0,1 & 0,1 \\ -0,2 & -0,1 \\ 0,1 & 0,3 \\ 0,15 & -0,2 \\ 0,0 & 0,0 \\ 0,0 & 0,0 \\ 0,0 & 0,0 \end{pmatrix}.$$

Примем матрицу коэффициентов выходного слоя НС

$$w_2 = \begin{pmatrix} 0,2 \\ 0,2 \\ 0,3 \end{pmatrix}.$$

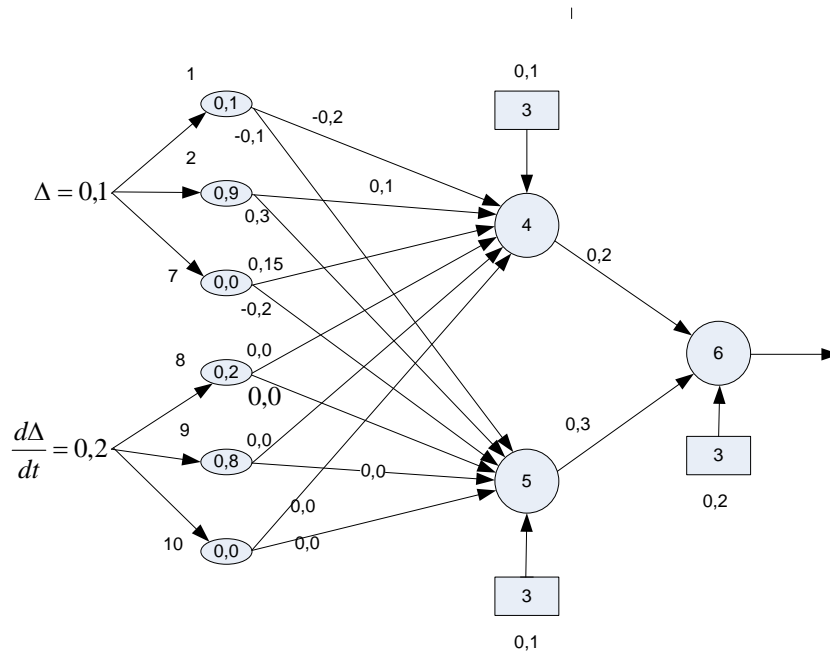


Рис.2-6. Исходное состояние ИНС

Примем начальные (целевые) значения степеней принадлежности терм

Н и ПМ для значений $\Delta = 0,1$ и $\frac{d\Delta}{dt} = 0,2$

$$X = \begin{pmatrix} 0,1 \\ 0,9 \\ 0,2 \\ 0,8 \end{pmatrix}.$$

Рассмотрим порядок расчета текущей эпохи ИНС

Определяем выходы элементов скрытого и выходного слоев по формулам:

$$net_j = \sum_{x=0} x_i w_{ij};$$

$$O_j = \frac{1}{1 + \exp(-net_j)}$$

Для нейрона 4

$$net_4 = (0,1 \times 1) + (0,1 \times -0,2) + (0,9 \times 0,1) = 0,17;$$

$$O_4 = \frac{1}{1 + \exp(-0,17)} = 0,542$$

Для нейрона 5

$$net_5 = (0,1 \times 1) + (0,9 \times 0,3) + (0,1 \times -0,1) = 0,36;$$

$$O_5 = \frac{1}{1 + \exp(-0,36)} = 0,589$$

Для нейрона 6

$$net_6 = (0,2 \times 1) + (0,542 \times 0,2) + (0,589 \times 0,3) = 0,485$$

$$O_6 = \frac{1}{1 + \exp(-0,485)} = 0,619$$

2. Проверяем разность между реальным и целевым выводами ИНС

$$(t_j - O_j) = (0,9 - 0,619) = 0,281;$$

$$0,01 \leq 0,281.$$

3. Определяем ошибку выходного нейрона 6 по формуле

$$\delta_j = (t_j - O_j) [O_j (1 - O_j)]$$

или

$$\delta_6 = \delta_k = (0,9 - 0,619) [0,619 (1 - 0,619)] = 0,06\epsilon.$$

4. Определяем ошибку нейронов 4 и 5 закрытого слоя по формуле

$$\delta_k = [(1 - O_k) O_k] \delta_j w_{kj},$$

где индекс к – соответствует предыдущему слою при обратном движении в ИНС.

5. Для нейрона 4

$$\delta_4 = [(1 - 0,542)0,542] \cdot 0,066 \cdot 0,2 = 0,003.$$

Для нейрона 5

$$\delta_5 = [(1 - 0,589)0,589] \cdot 0,066 \cdot 0,3 = 0,005.$$

6. Определяем для всех слоев приращение значений весов каждого нейрона по формуле

$$\Delta w_{ij} = \eta \cdot \delta_j \cdot O_j$$

где η – коэффициент скорости обучения; $\eta = 0,01 \div 1,0$.

Для нейрона 6

$$\Delta w_{46} = 0,25 \cdot 0,066 \cdot 0,542 = 0,009;$$

$$\Delta w_{56} = 0,25 \cdot 0,066 \cdot 0,589 = 0,01;$$

$$\Delta w_{36} = 0,25 \cdot 0,066 \cdot 1 = 0,017.$$

Для нейрона 4

$$\Delta w_{84} = \Delta w_{94} = \Delta w_{24} = \Delta w_{14} = 0,25 \cdot 0,003 \cdot 0,542 = 0,0004;$$

$$\Delta w_{34} = 0,25 \cdot 0,005 \cdot 1 = 0,00125.$$

Для нейрона 5

$$\Delta w_{95} = \Delta w_{85} = \Delta w_{25} = \Delta w_{15} = 0,25 \cdot 0,005 \cdot 0,585 = 0,00073;$$

$$\Delta w_{35} = 0,25 \cdot 0,003 \cdot 1 = 0,00075.$$

7. Определяем значения коэффициентов каждого элемента ИНС для выполнения расчетов следующей эпохи по формуле

$$w_{ij}^1 = w_{ij} + \Delta w_{ij}.$$

Для нейрона 6

$$w_{46}^1 = w_{46} + \Delta w_{46} = 0,3 + 0,01 = 0,31;$$

$$w_{56}^1 = w_{56} + \Delta w_{56} = 0,2 + 0,009 = 0,209;$$

$$w_{36}^1 = w_{36} + \Delta w_{36} = 0,2 + 0,017 = 0,217.$$

Для нейрона 4

$$w_{14}^1 = w_{14} + \Delta w_{14} = -0,2 + 0,0004 = -0,1996;$$

$$w_{24}^1 = w_{24} + \Delta w_{24} = 0,1 + 0,0004 = 0,1004;$$

$$w_{84}^1 = w_{84} + \Delta w_{84} = 0,0 + 0,0004 = 0,0004$$

$$w_{94}^1 = w_{94} + \Delta w_{94} = 0,0 + 0,0004 = 0,0004$$

$$w_{34}^1 = w_{34} + \Delta w_{34} = 0,1 + 0,00125 = 0,10125$$

Для нейрона 5

$$w_{15}^1 = w_{15} + \Delta w_{15} = -0,1 + 0,00073 = -0,09927;$$

$$w_{25}^1 = w_{25} + \Delta w_{25} = 0,3 + 0,00073 = 0,30073;$$

$$w_{85}^1 = w_{85} + \Delta w_{85} = 0,0 + 0,00073 = 0,00073;$$

$$w_{95}^1 = w_{95} + \Delta w_{95} = 0,0 + 0,00073 = 0,00073$$

$$w_{35}^1 = w_{35} + \Delta w_{35} = 0,1 + 0,00075 = 0,100075$$

8. Запишем матрицу коэффициентов для расчета ИНС последующей эпохи.

$$w_1 = \begin{array}{|cc} 0,10125 & 0,100075 \\ -0,1996 & -0,09927 \\ 0,1004 & 0,30073 \\ 0,15 & -0,2 \\ 0,0004 & 0,0004 \\ 0,00073 & 0,00073 \\ 0,0 & 0,0 \end{array};$$

$$w_2 = \begin{array}{|c} 0,217 \\ 0,209 \\ 0,310 \end{array}.$$

На рис. 2-7 показано состояние ИНС после первой эпохи.

Для дальнейшего обучения ИНС необходимо определить статическую регулировочную характеристику $\mu = f(\Delta\sigma)$ конкретного РО. Число учебных образцов N для получения допустимого значения ошибки в процессе тестирования определяется по формуле

$$N \geq \frac{W}{a},$$

где W – число весовых коэффициентов ИНС. Так, при допустимости 10% ошибок число учебных образцов должно быть в 10 раз больше числа имеющихся в сети весовых коэффициентов. В ИНС участвует семнадцать

весовых коэффициентов и для получения 10% ошибки обучения необходимо иметь 170 учебных образцов.

В процессе обучения нейро-контроллер запомнит параметры матриц 170 учебных образцов. При включении нейронного регулятора в работу в зависимости от значений ошибки отклонения и ее производной нейро-контроллер выберет необходимые матрицы НС и выдаст сигнал управления на вход регулируемого объекта (РО).

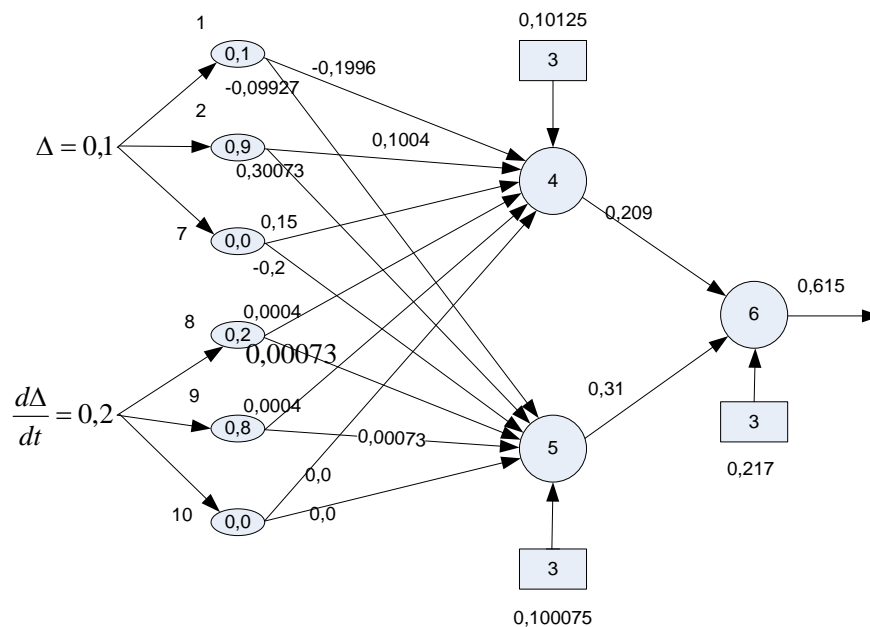


Рис. 2-7. Состояние ИНС после первой эпохи

При этом нейронный регулятор будет изменять сигнал управления с целью доведения ошибки отклонения и ее производной до нуля по методу ОРО с помощью программы *BreinMaker* [14]. В этом заключается адаптивность нейронного регулятора, которая придает регулятору квазиастатические свойства.

Обычно применяется комбинация градиентного спуска в виде ОРО и МНК. По МНК оцениваются коэффициенты заключений правил, так как они линейно связаны с выходом сети. Каждая итерация процедуры настройки выполняется в два этапа. На первом этапе на входы подается обучающая выборка и по невязке между желаемым и действительным поведением сети МНК находятся оптимальные параметры узлов четвертого слоя. На втором

этапе остаточная невязка передается с выхода сети на входы и методом ОРО модифицируются параметры узлов первого слоя. При этом найденные на предыдущем этапе коэффициенты заключений правил не изменяются. Итерационная процедура настройки продолжается, пока невязка превышает заранее установленное значение. Для настройки ФП, кроме метода ОРО, могут использовать и другие алгоритмы оптимизации, например, алгоритм Левенберга-Марквардта – алгоритм нелинейной оптимизации, использующий для поиска минимума комбинированную стратегию – линейную аппроксимацию и градиентный спуск. Переключение с одной стратегии на другую происходит в зависимости от того, была ли успешной линейная аппроксимация. Такой подход называют моделью доверительных областей.

Необходимо обратить внимание на то, что для уменьшения времени настройки весов нейронного регулятора возможно применение генетического алгоритма (ГА), который с помощью программы *Evolver* найдет начальные значения весов, достаточно близкие к оптимальным значениям [15].

Этот алгоритм есть обобщение дельта-метода, осуществляемого послойно, от последнего (выходного) слоя к входному слою. Вычислительная сложность при адаптации МНС снижается распараллеливанием каналов обработки информации и не требует знания о нелинейностях и параметрах объекта. Для проведения обучения МНС требуется знание о поведении регулируемого объекта в каких-то эталонных режимах или наличие эталонной модели, задающей желаемое движение в системе.

2.5.2. Генетические алгоритмы

Недостатком нейронных сетей является значительное время на ее обучение. Это связано с большим размером матрицы весов НС и выбором их начальных значений. Задача генетического алгоритма (ГА) оптимизировать

поиск начальных значений матрицы весов и этим существенно сократить время обучения НС [4].

В общем случае ГА – стохастические, эвристические и оптимизационные методы, которые основаны на идее эволюции с помощью естественного отбора, выдвинутой Дарвиным (1857 г.). ГА, впервые предложенные через 118 лет Дж. Холландом в 1975 г. Есть вычислительная технология, которая используется в системах искусственного интеллекта, оптимизации и программного обеспечения. ГА – простая модель эволюции в природе, реализованная в виде компьютерных программ.

Эволюция – процесс оптимизации всех живых организмов. Механизм эволюции – естественный отбор.

Естественный отбор – процесс формирования новой популяции из старой популяции, которая в дальнейшем погибает. В природе наблюдается быстрая обработка информации в нервной системе и медленный процесс естественной эволюции. Современные ЭВМ снимают это противоречие, т.е. ускоряют процесс эволюции.

Популяция – конечное множество особей (организмов).

Функция приспособления (функция оценки) указывает меру приспособления данной особи в популяции. В задачах оптимизации ее называют целевой функцией.

Хромосома – дезоксирибонуклеиновая кислота (ДНК) в оболочке или упорядоченная последовательность генов. В дальнейшем хромосома будет представляться как вектор (последовательность) из нулей и единиц.

ДНК – цепочка, состоящая из молекул нуклеотидов четырех типов (F, N, C, G), которые располагаясь особым образом, определяют код индивидуума.

Ген – определенная часть хромосомы.

Клетки человека имеют 46 хромосом. Длина хромосомы у человека может состоять из 100000 генов, хотя их точное число неизвестно. Каждое врожденное качество особи (цвет глаз, наследственные болезни и т. д.)

кодируются определенной частью хромосомы. Например, ген цвет глаз содержит информацию, кодирующую цвет глаз.

Аллель – различные значения гена.

Генотип – набор хромосом данной особи.

Фенотип – набор значений, соответствующих данному генотипу.

Локки – множество позиций генов.

Кроссовер – скрещивание ДНК предков и образование ДНК потомка.

Мутация – случайное (стохастическое) изменение одной или нескольких позиций в хромосоме под действием радиации или других влияний.

Возможно независимое применение варианта ГА и НС для решения одной и той же задачи, показанной на рис. 2-8, а также вспомогательное применение варианта ГА и НС, когда они дополняют друг друга, и показанного на рис.2-9.

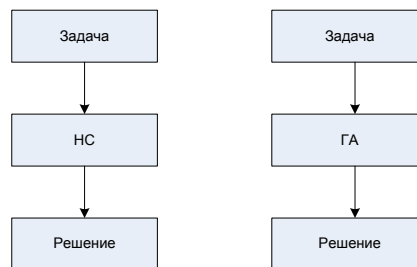


Рис.2-8. Варианты независимых вычислительных технологий

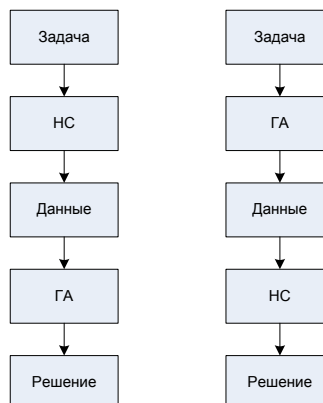


Рис.2-9. Варианты вспомогательных объединений вычислительных технологий

В первом варианте НС предназначена для поддержки ГА. НС формирует исходную популяцию для ГА.

Во втором варианте ГА предназначена для поддержки НС. При этом возможно решение трех задач: применение варианта ГА для подбора весов НС гибридным методом; для подбора правил обучения либо параметров, управляющих обучением НС; для анализа НС с целью понимания, что и почему делает НС.

Гибридный подход состоит в объединении методов ГА и ОРО для подбора весов НС. Как правило, вначале при помощи ГА находится решение (начальное значение весов), достаточно близкие к оптимальным значениям с помощью программы *Evolver*, и затем оно рассматривается как отправная точка для традиционного поиска весов, например, ОРО с помощью программы *Brein Maker*.

Рассмотрим НС, показанную на рис. 2-10 как отправная точка для традиционного поиска оптимальной точки весов, например, ОРО с помощью программы *Brein Maker*.

Выбор оптимального первоначального набора весов НС выполним с помощью ГА (программа *Evolver*) [34], приведенную на рис.2-10.

На рис. 2-10 показаны девять параметров (хромосом), которые определяют точку в пространстве поиска и представляют собой возможное решение.

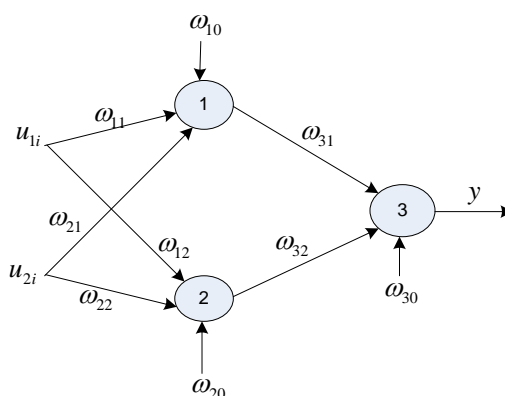


Рис.2-10. Нейронная сеть

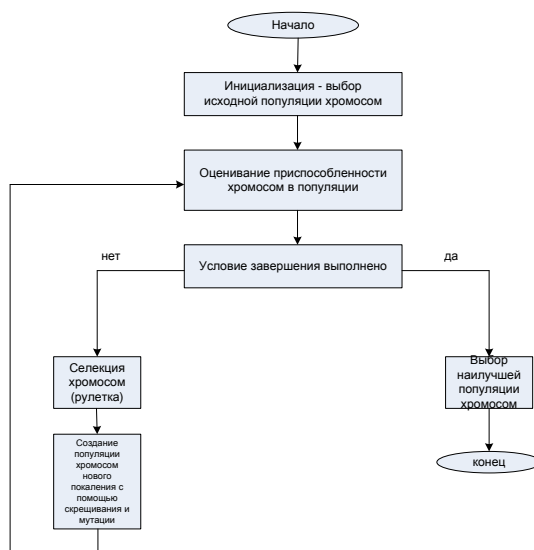


Рис.2-11. Блок-схема генетического алгоритма

Допустим, что веса могут принимать значения в интервале от -10 до $+10$.

Для этого необходимо обеспечить:

- кодирование хромосом табличным методом;
- определение функции приспособленности хромосом;
- расчет выхода НС из начальных значений хромосом;
- селекция начальных хромосом методом рулетки;
- формирование новой популяции хромосом с помощью оператора «скрещивание»;
- кодирование новой популяции хромосом;
- определение функций приспособленности новой популяции хромосом;
- расчет выхода НС с новыми значениями хромосом;
- сравнение значения старого выхода НС с новым значением выхода НС и т.д.

Рассмотрим подробно работу ГА.

Пример 2-2. Найти хромосому с максимальным количеством единиц, где хромосомы состоят из 12 генов, а популяция насчитывает 8 хромосом. Тогда лучшая хромосома должна содержать 12 единиц.

Инициализация

Необходимо случайным образом сгенерировать 8 двоичных последовательностей (хромосом) длиной 12 бит. Этого можно достигнуть, например, подбрасыванием монет $8 \times 12 = 96$ раз, где при выпадении «орла» – 1, выпадении «решка» – 0. Таким образом, сформируем исходную популяцию.

$$Ch_1 = [11100110001];$$

$$Ch_2 = [00110011100];$$

$$Ch_3 = [01110111001];$$

$$Ch_4 = [00100010100];$$

$$Ch_5 = [01000110000];$$

$$Ch_6 = [01001100001];$$

$$Ch_7 = [10101101101];$$

$$Ch_8 = [0000101100].$$

Оценка приспособленности хромосом в популяции

Функция приспособленности F определяется количеством единиц в хромосоме. Тогда оценка приспособленности для каждой хромосомы в исходной популяции будет такой:

$$F(Ch_1) = 7;$$

$$F(Ch_2) = 6;$$

$$F(Ch_3) = 8;$$

$$F(Ch_4) = 3;$$

$$F(Ch_5) = 4;$$

$$F(Ch_6) = 5;$$

$$F(Ch_7) = 8;$$

$$F(Ch_8) = 5.$$

Хромосомы Ch_3 и Ch_7 имеют наибольшее значение функции приспособленности и являются лучшими кандидатами на решение задачи. Если в соответствии с блок-схемой ГА условие остановки не выполняется, то на следующем шаге ГА производится селекция хромосом из текущей популяции с помощью рулетки или турнирного метода.

Селекция

Селекция хромосом с помощью рулетки проводится на основании формулы

$$v(Ch_i) = \frac{F(Ch_i)}{\sum_{i=1}^N F(Ch_i)} 100\% .$$

Селекция хромосом может быть представлена как результат поворота рулетки, поскольку «выигрышная» хромосома относится к выпавшему сектору этого колеса. Чем больше сектор, тем больше вероятность победы соответствующей хромосомы. Вероятность выбора данной хромосомы оказывается пропорциональной значению ее функции приспособленности. Выбор с помощью колеса рулетки сводится к выбору числа из интервала $[0...100]$, которое соответствует конкретной точке на окружности колеса. В результате селекции создается родительская популяция (родительский пул) с численностью $N = 8$, равной текущей популяции.

Метод рулетки отбирает особи с помощью $N = 8$ «запусков» рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер i -сектора пропорционален соответствующей величине $v(Ch_i)$. При таком отборе члены популяции, имеющие более высокую приспособленность, выбираются будут чаще, чем особи с низкой приспособленностью.

Для нашего примера получим:

$$v(Ch_1) = 15,22\% ;$$

$$v(Ch_2) = 13,04\% ;$$

$$\nu(Ch_3) = 17,39\%;$$

$$\nu(Ch_4) = 6,52\% ;$$

$$\nu(Ch_5) = 8,7\%;$$

$$\nu(Ch_6) = 10,87\%;$$

$$\nu(Ch_7) = 17,39\%;$$

$$\nu(Ch_8) = 10,87\%.$$

На рис. 2-12 показано колесо рулетки.

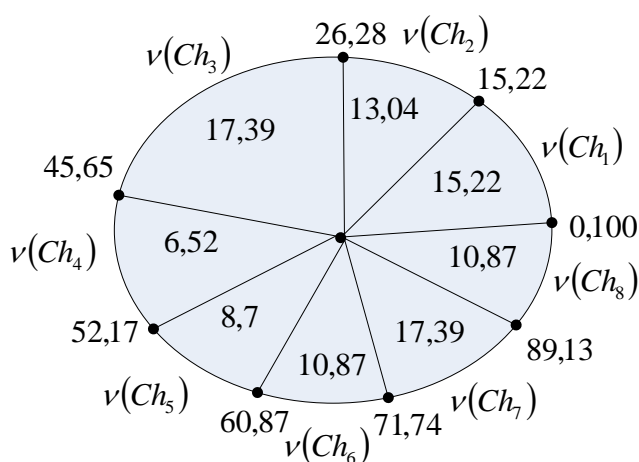


Рис. 2-12. Колесо рулетки

Селекция хромосом турнирного метода реализуется N турнирами для выбора N особей. Каждый турнир построен на выборе k элементов из популяции и выборе лучшей особи из них. Обычно $k = 2$.

1. Выбор новой популяции особей с помощью скрещивания и мутации.

Оператор «скрещивание» осуществляет обмен частями хромосом между двумя (может быть и больше) хромосомами. При одноточечном скрещивании случайным образом выбирается одна из $l-1$ -точек разрыва. Обе родительские структуры разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных родителей склеиваются и получают два генотипа потомков. На рис.2-22 показан одноточечный оператор скрещивания (точка разрыва 3).

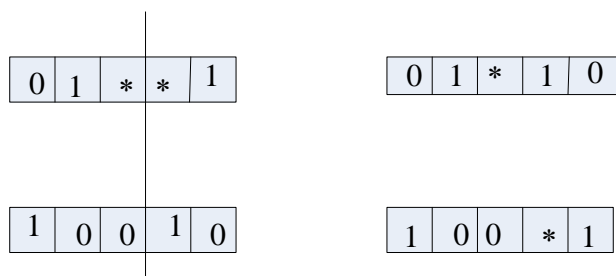


Рис.2-13. Одноточечный оператор скрещивания

При многоточечном скрещивании точек разрыва может быть больше.

Пример генетического алгоритма

Начало (генетический алгоритм);

Создать начальную популяцию;

Оценить приспособленность каждой точки (хромосомы);

Если популяция не сошлась, то окончанию ГА – нет;

Начало (создать популяцию нового поколения);

Выполнить селекцию (размер популяции /2) раз;

Начало (цикл воспроизводства);

Выбрать две особи с высокой приспособленностью из предыдущего поколения;

Скрестить выбранные особи и получить двух потомков;

Оценить приспособленность потомков;

Поместить потомков в новое поколение;

Конец;

Если популяция сошлась, то окончанию ГА – да;

Конец;

Конец.

2.6. Адаптивный нейрон. Алгоритмы адаптации

Рассмотрим нечеткое управление с применением нейронной технологии. Постоянно растущая сложность и разнообразие задач, решение которых возлагается на автоматические системы, в последнее время

определяют повышенную потребность в системах управления с более универсальными свойствами. Подходящим для решения большинства задач являются системы, основанные на нечеткой логике и ИНС [16]. Нейронные сети выполняют решения, используя предшествующий опыт, что дает возможность приспособиться к изменениям в системе. Объединение элементов нечеткой логики (фаззификация) и НС позволит исключить недостатки нечеткой логики и НС и проектировать адаптивные нечеткие регуляторы.

По сравнению с традиционными методами анализа и вероятностным подходом нечеткое управление с применением нейронной технологии позволяет проводить анализ задачи и получать результаты с заданной точностью, обеспечивать значительное повышение быстродействия процессов управления при использовании нейро-нечетких контроллеров и создания систем управления для объектов, где нецелесообразно применять методы традиционной математики. Мощь НС как методологии разрешения проблем гарантирует ее успешное использование в нечетких регуляторах.

Нечеткое регулирование с применением нейронной технологии повышает быстродействие, точность и качество регулирования. Позволяет работать с контурами, которыми раньше можно было управлять только вручную, в которых возможно лишь нелинейное регулирование.

Регулирование на основе нечеткой логики с применением ИНС на базе адаптивных нейронов является простым и надежным в реализации.

На рис. 2-14 в качестве примера приведен адаптивный фаззофикатор.

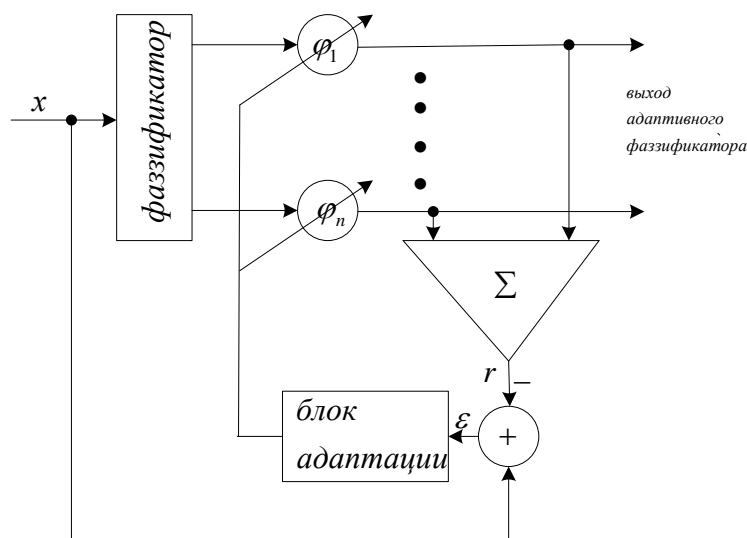


Рис.2-14. Адаптивный фаззофикатор

В качестве алгоритма адаптации могут быть градиентные алгоритмы:

- итерационный алгоритм адаптации Уидроу-Хоффа с переменным шагом;
- модифицированный метод наименьших квадратов;
- метод последовательного обучения.

Алгоритм адаптации Уидроу-Хоффа

Уидроу и Хофф модифицировали персептронный алгоритм Ф. Розенблатта, дополнительно введя сигмоидальную функцию активации [6].

Их модели Адлин (с одним выходным нейроном) и Мадалин (много выходных нейронов) получили широкое распространение. Они доказали, что сеть при определенных условиях будет сходиться к любой функции, которую она может представить. Процедура Уидроу-Хоффа разработана применительно к «черному ящику», в котором между входами и выходами существуют только прямые связи. Процедура обучения Мадалин состоит в том, что веса внутренних связей между нейронами подстраиваются до тех пор, пока не установится требуемое соотношение между входными и выходными векторами. Процесс состоит из двух чередующихся фаз. В первой фазе на входах задается входной вектор, а на выходах – нужный выходной вектор. Затем веса всех связей, соединяющих активные входы и выходы,

увеличиваются на малую величину δ . Во второй фазе на входе формируется тот же входной вектор, однако теперь «черный ящик» решает, какой вектор сформировать на выходе. При этом должно соблюдаться следующее правило: выход активизируется только тогда, когда сумма весов его связей с активными входами положительна. После этого веса всех связей, соединяющих активные входные и выходные элементы, уменьшаются на величину δ . Если сеть выработала правильный выходной вектор, то эти уменьшения весов в точности компенсируют их увеличение, произведенные в первой фазе, поскольку в обеих фазах активны одни и те же пары вход-выход. Если же сеть выработала не тот выходной вектор, который нужен, то изменение весов, рассчитанных в первой фазе, сохраняется. При реализации алгоритма обучения персептрона Адалин появляются отличие от классического персептронного алгоритма в четвертом шаге, где используются непрерывные сигналы вместо бинарных сигналов. Недостатком процедуры Уидроу-Хоффа является требования наличия в «черном ящике» промежуточных слоев, которые бы выделяли из входного вектора иерархию тех его признаков, которые и определяли бы в конечном итоге выбор выходного вектора. На практике с целью обучения нейрона его охватывают обратной связью через блок алгоритма адаптации.

Элемент сравнения в цепи обратной связи сравнивает фактический сигнал с выхода сумматора нейрона с желаемым сигналом d (это может быть ошибка регулирования в САР или производная ошибки регулирования). Алгоритм адаптации подстраивает коэффициенты вектора входных сигналов так, чтобы свести к нулю ошибку $\varepsilon = (d - r)$. Процедурно это организуется сведением к минимуму квадрата ошибки

$$\varepsilon^2 = (d - r)^2, \quad (2-1)$$

где

$$r = \sum_{i=1}^n w_i x_i. \quad (2-2)$$

Для решения данной безусловной оптимизационной задачи используем градиентный метод (дельта-метод). В этом методе следующая $(k+1)$ -итерация для значения i -го весового коэффициента w_i находится по формуле

$$w_i(k+1) = w_i(k) - \left(\frac{d(\varepsilon)^2}{dw_i} \right)_k C, \quad (2-3)$$

где C – положительная константа.

Из (2-1) с учетом (2-2) следует, что

$$\frac{d(\varepsilon)^2}{dw_i} = 2\varepsilon \left(-\frac{dr}{dw_i} \right) = -2\varepsilon x_i. \quad (2-4)$$

Тогда окончательное выражение для алгоритма адаптации будет иметь вид

$$w_i(k+1) = w_i(k) + 2C\varepsilon(k)x_i(k), \quad (2-5)$$

где $2C$ – скорость обучения нейрона, определяемая скоростью сходимости итерационного процесса оптимизации.

Суть адаптации нейрона заключается в следующем.

На базе адаптивного нейрона создана следящая система, заданием которой является случайная величина, например, отклонение напряжения от заданного значения. Задача следящей системы, меняя коэффициенты (синапсы) нейрона, обрабатывать отклонения между выходом сумматора соответствующего нейрона и заданием его следящей системы с помощью итерационной процедуры с переменным шагом (алгоритм адаптации Уидроу-Хоффа). Алгоритм адаптации подстраивает коэффициенты входной матрицы так, чтобы свести к нулю ошибку ε . Процедурно это организуется сведением к минимуму квадрата ошибки. Воздействие выхода адаптивного нейрона через активационную функцию F , парирует текущее отклонение от заданного значения с учетом инерционности системы. В процессе адаптации происходит коррекция синапсов адаптивного нейрона до тех пор, пока

желаемый сигнал Δ не стане равным нулю, т.е. нейрон исключая ошибку в статике, полностью адаптировался.

Достоинством алгоритма Уидроу-Хоффа является отсутствия ограничения на вид функции принадлежности (терма) и ее расположение в нормированном интервале, а недостаток – сравнительно низкое быстродействие.

Модифицированный метод наименьших квадратов

Для повышения быстродействия рассмотрим алгоритм адаптации, основанный на модификации метода наименьших квадратов [8]. Данный алгоритм в отличие от алгоритма Уидроу-Хоффа рассчитывает синаптические веса W_{kj} нейрона за одну итерацию из условия линейных терм-множеств фаззификатора лингвистической переменной (например, отклонение текущего параметра).

Пример 2-3. Вывести формулы расчета синапсов согласно модифицированному методу наименьших квадратов.

1. Выбираем терм-множество из линейных функций принадлежности в нормированном интервале $-1 \dots +1$.
2. Разбиваем нормированный интервал, например, на пять подинтервалов (рис. 2.15):

$$Y_1 \in [-1; -0,6];$$

$$Y_1 \in [-0,6; -0,2];$$

$$Y_1 \in [-0,2; +0,2];$$

$$Y_1 \in [+0,2; +0,6];$$

$$Y_1 \in [+0,6; +1].$$

3. Записывает функционал согласно методу наименьших квадратов для данного примера

$$F = (W_1 X_1 + W_2 X_2 + W_3 X_3 + W_4 X_4 + W_5 X_5 - Y_i)^2$$

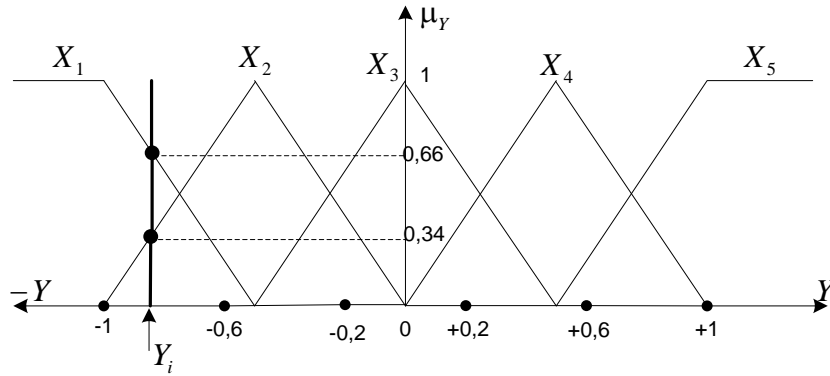


Рис. 2-15. К разбиению на подинтервалы нормированного интервала

4. Взяв производную от функции F по каждому синапсу на каждом из 5 интервалов, и приравняв их к нулю, получим:

$$\frac{\partial F}{\partial W_1} = 2(W_1 X_1 + W_2 X_2 + W_3 * 0 + W_4 * 0 + W_5 * 0 - Y_i) = 0;$$

$$\frac{\partial F}{\partial W_2} = 2(W_1 X_1 + W_2 X_2 + W_3 * X_3 + W_4 * 0 + W_5 * 0 - Y_i) = 0;$$

$$\frac{\partial F}{\partial W_3} = 2(W_1 * 0 + W_2 X_2 + W_3 X_3 + W_4 * X_4 + W_5 * 0 - Y_i) = 0;$$

$$\frac{\partial F}{\partial W_4} = 2(W_1 * 0 + W_2 * 0 + W_3 X_3 + W_4 X_4 + W_5 X_5 - Y_i) = 0;$$

$$\frac{\partial F}{\partial W_5} = 2(W_1 * 0 + W_2 * 0 + W_3 * 0 + W_4 X_4 + W_5 X_5 - Y_i) = 0.$$

5. После преобразований получим:

$$\frac{\partial F}{\partial W_1} = W_1 X_1 + W_2 X_2 - Y_i = 0;$$

$$\frac{\partial F}{\partial W_2} = W_1 X_1 + W_2 X_2 + W_3 * X_3 - Y_i = 0;$$

$$\frac{\partial F}{\partial W_3} = W_2 X_2 + W_3 X_3 + W_4 * X_4 - Y_i = 0;$$

$$\frac{\partial F}{\partial W_4} = W_3 X_3 + W_4 X_4 + W_5 X_5 - Y_i = 0;$$

$$\frac{\partial F}{\partial W_5} = W_4 X_4 + W_5 X_5 - Y_i = 0.$$

6. Выразим синапсы (весовые коэффициенты) следующим образом:

$$\begin{aligned}
W_1 &= \frac{Y_i - W_2 X_2}{X_1}; \\
W_2 &= \frac{Y_i - W_1 X_1 - W_3 X_3}{X_2}; \\
W_3 &= \frac{Y_i - W_2 X_2 - W_4 X_4}{X_3}; \\
W_4 &= \frac{Y_i - W_3 X_3 - W_5 X_5}{X_4}; \\
W_5 &= \frac{Y_i - W_4 X_4}{X_5}.
\end{aligned}
\tag{2-6}$$

В формулах (2-6) Y_i – есть текущее значение переменной величины, подаваемое на вход фаззификатора и в тоже время есть задание следящей системы, которые должны изменяться синхронно.

При первом отсчете начальные значения синапсов задаются случайным образом, а в дальнейшем за исходные принимаются значения синапсов предыдущего отсчета. При этом каждый раз значения синапсов (в зависимости от того какому интервалу принадлежит желаемый Y) автоматически подстраиваются согласно (2-6) под новые значения. В случае, если отклонение вышло за нормированный диапазон, то $W_1 = Y_1$ и $W_5 = Y_1$.

Модифицированный алгоритм наименьших квадратов применим только для линейных терм, поддиапазоны которых равны и равномерно расположены в нормированном интервале.

Число терм терм-множества фаззификатора всегда должно быть равно числу подинтервалов разбиения нормированного интервала. В случае включения регулятора терм – модификация метода наименьших квадратов не допустима. В дальнейшем предлагается метод адаптации с минимальными итерациями или метод с последовательным обучением (рассмотренный выше).

Алгоритм последовательного обучения

Рассмотрим метод адаптации с последовательным обучением [17]. Данный метод относится к методам последовательной идентификации линейных систем в реальном времени, который основан на принципе обучения с моделью и в котором получена модель импульсной характеристики. Аналогично методам стохастической аппроксимации метод последовательного обучения не дает оценки параметров по методу наименьших квадратов на различных последовательных шагах, вследствие чего сходимость этого метода несколько ниже, чем метод последовательной регрессии. При этом оценки постепенно сходятся (в среднем) к истинным значениям параметров. Метод последовательного обучения отличается от метода стохастической аппроксимации характеристиками сходимости и его удобно применять для процессов с медленно изменяющимися параметрами. Основное преимущество метода последовательного обучения состоит в простоте алгоритма адаптации.

В методе последовательного обучения рассматривается линейная система со случайным входом $u(t)$, выходом $x(t)$ и импульсной характеристикой $g(t)$. Связь между входом и выходом описывается интегралом свертки, который для нулевых начальных условий записывается в виде

$$x(t) = \int_0^t g(\tau)u(t-\tau)d\tau, \quad (2-7)$$

где $u(t-\tau)$ – измеряемая величина.

Дискретная форма интеграла свертки

$$x_j = \sum_{i=1}^N g_i u_{j-i}. \quad (2-8)$$

Определение импульсной реакции g_i выполняется путем итерационных вычислений множества величин: $\phi_1^{(j)}, \phi_2^{(j)}, \dots, \phi_N^{(j)}, \forall j = N+1, N+2, \dots$, которые должны, соответственно, приближаться к g_1, \dots, g_N в уравнении (2-8),

где j обозначает номер итерации. Тогда оценка выхода модели (нейрона) x_{jm} при использовании $\phi_i^{(j)}$ равна по аналогии с (2-9):

$$x_{jm} = \sum_{i=1}^N \phi_i^{(j)} u_{j-i} . \quad (2-9)$$

Обозначая

$$\begin{aligned} g &= [g_1, \dots, g_N]^T ; \\ \phi_j &= [\phi_1(j), \dots, \phi_N(j)]^T ; \\ u_j &= [u_{j-1}, \dots, u_{j-N}]^T . \end{aligned}$$

Преобразуем (2-8) и (2-9), соответственно, к виду

$$\begin{aligned} x_j &= g^T u_j ; \\ x_{jm} &= \phi_j^T u_j . \end{aligned}$$

Определяем разность

$$x_j - x_{jm} = (g - \phi_j)^T u_j = u_j^T (g - \phi_j) = u_j^T \Delta \phi_j .$$

Рассмотрим вектор $\Delta \phi = \phi_{j+1} - \phi_j$, который будем использовать для коррекции вектора следующей идентификацией ϕ_{j+1} относительно ϕ_j с учетом ошибки $(x_j - x_{jm})$ в оценке x_j .

Полагая $\Delta \phi_j = (x_j - x_{jm}) \frac{u_j}{u_j^T u_j}$, где $j = 1, 2, \dots$, получим рекуррентную формулу

$$\phi_{j+1} = \phi_j + (x_j - x_{jm}) \frac{u_j}{u_j^T u_j} . \quad (2-10)$$

Чтобы начать процедуру оценивания ошибки согласно (2-10) при $j=1$ можно подставить $x_{jm} = 0$.

Если в (2-10) ввести коэффициент коррекции ошибки C , то получим окончательное выражение рекуррентной формулы:

$$\phi_{j+1} = \phi_j + C(x_j - x_{jm}) \frac{u_j}{u_j^T u_j} , \text{ при } 0 < C < 2 . \quad (2-11)$$

Пример 2-4. Рассмотреть применение рекуррентной формулы (2-11) для адаптации искусственного нейрона.

Пусть терм-множество фаззификатора имеет вид, где при текущем отклонении $\Delta u_j = 0,4$ активизировались термы *Н* и *П* с формированием степеней принадлежности $\mu_{ПБ} = 0,4$ и $\mu_{ПС} = 0,6$.

Запишем (2-11) в новых обозначениях при условии $C = 1$

$$\phi_{j+1} = \phi_j + \frac{C \varepsilon_j \begin{bmatrix} \mu_{ПБ} \\ \mu_{ПС} \end{bmatrix}}{\mu_{ОБ}^2 + \mu_{ОС}^2 + \mu_{ПС}^2 + \mu_{ПБ}^2},$$

где $\varepsilon = \Delta u_j - (\phi_{ПБ} \mu_{ПБ} + \phi_{ПС} \mu_{ПС})$.

Расчет первой итерации

$$\begin{bmatrix} \phi_{ПБ}^{(1)} \\ \phi_{ПС}^{(1)} \end{bmatrix} = \begin{bmatrix} \phi_{ПБ}^{(0)} \\ \phi_{ПС}^{(0)} \end{bmatrix} + \frac{(\Delta u_j - \phi_{ПБ}^{(0)} \cdot \mu_{ПБ} - \phi_{ПС}^{(0)} \cdot \mu_{ПС}) \begin{bmatrix} \mu_{ПБ} \\ \mu_{ПС} \end{bmatrix}}{[\mu_{ПБ}]^2 + [\mu_{ПС}]^2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{(0,4) \begin{bmatrix} 0,4 \\ 0,6 \end{bmatrix}}{0,4^2 + 0,6^2} = \begin{bmatrix} 0,3077 \\ 0,4615 \end{bmatrix}.$$

Расчет второй итерации

$$\begin{bmatrix} \phi_{ПБ}^{(2)} \\ \phi_{ПС}^{(2)} \end{bmatrix} = \begin{bmatrix} \phi_{ПБ}^{(1)} \\ \phi_{ПС}^{(1)} \end{bmatrix} + \frac{(\Delta u_j - \phi_{ПБ}^{(1)} \cdot \mu_{ПБ} - \phi_{ПС}^{(1)} \cdot \mu_{ПС}) \begin{bmatrix} \mu_{ПБ} \\ \mu_{ПС} \end{bmatrix}}{\mu_{ПБ}^2 + \mu_{ПС}^2} = \begin{bmatrix} 0,3077 \\ 0,4615 \end{bmatrix} + \frac{(0,4 - 0,3077 \cdot 0,4 - 0,4615 \cdot 0,6) \cdot \begin{bmatrix} 0,4 \\ 0,6 \end{bmatrix}}{0,4^2 + 0,6^2} =$$

$$\rightarrow \begin{bmatrix} 0,3077 \\ 0,4615 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,3077 \\ 0,4615 \end{bmatrix}.$$

Таким образом, потребовалась одна итерация на данном шаге, вторая итерация приведена только для доказательства, подтверждающая повторяемость результата.

Из анализа рассмотренных алгоритмов предпочтение следует отдать методу с последовательным обучением за его быстродействие и возможность использования как линейных, так и нелинейных терм.

Сигнал с выхода адаптивного нейрона подается на блок активационной функции для дальнейшего преобразования.

Из анализа рассмотренных алгоритмов предпочтение следует отдать методу с последовательным обучением за его быстродействие и возможность использования как линейных, так и нелинейных терм.

В настоящее время наблюдается интенсивное развитие и практическое применение адаптивных систем для целей управления и регулирования многих технических объектов, в частности авиационного двигателя.

Эффективность любого летательного аппарата зависит от совершенства системы автоматического управления силовой установки. Повышение надежности эксплуатации, экономии топлива, снижение эксплуатационных затрат силовых установок на базе турбовального турбореактивного двухконтурного двигателя (ТРДД), снижении их массы и стоимости делают актуальным создание комплексных систем управления, интегрируемых по аппаратуре и по способам управления двигателем.

Актуальность разработки адаптивных регуляторов с применением нейронной технологии в условиях неопределенности математического описания авиационного двигателя определяется следующим:

- при проектировании регуляторов авиационного двигателя разработчики не учитывают индивидуальные характеристики двигателя, а ориентируются на среднестатистическую статическую характеристику, которая задана в виде приближенной передаточной функции с переменными параметрами;

- не учитываются неконтролируемые возмущения, возникающие как в объекте управления (авиационном двигателе), так и в исполнительных механизмах (например, неконтролируемое изменение расхода топлива в дозаторе);

- явно нелинейный объект (авиационного двигателя с некорректным математическим описанием) управляется классическими (П-, ПИ-, ПИД-) регуляторами с набором настроек, заданных жестким массивом;

- отсутствие адаптации к переменным внешним условиям исключает экономный расход топлива и тем самым исключает увеличение дальности полета летательного аппарата.

В настоящее время известны следующие методы адаптации САУ при отсутствии математического описания объекта:

- беспоисковые прямые адаптивные системы;

- метод активной адаптации (гибкая эталонная модель);
- нечеткие нейронные системы регулирования с применением искусственных нейронов с последовательным обучением [17];
- нейронные сети прямого распространения (сети *ANFIS*, *TSK*, Ванга-Менделя и т.д.) [5];
- модифицированные нейронные сети прямого распространения.

Пример 2-5. Рассмотреть методику проектирования адаптивных регуляторов авиационного двигателя в условиях неопределенности его математического описания с применением метода адаптации с последовательным обучением.

Специфика проектирования системы управления авиационного двигателя требует выбора регулирующего органа (дозатора), датчика температуры за турбиной низкого давления, датчика частоты вращения ТРДД и т.д.

Рассмотрим возможные варианты решения данной задачи.

В качестве выбора дозатора с обратной связью управление последним возможно с помощью ПИ-регулятора, где интегральная составляющая закона регулирования запоминает текущее значение подачи топлива от выбранного селектором контура управления.

В случае выбора дозатора без обратной связи (неисправность, которого определяется неисправностью датчика обратной связи) для управления возможно применение ПД-закона регулирования, где интеграл, связанный с перемещением дозирующей иглы также запоминает текущую подачу топлива от выбранного селектором контура управления. При этом хорошо выполняется согласование между контурами управления авиационного двигателя.

В качестве датчика температуры возможно использование батареи термпар, установленных за турбиной низкого давления.

Учитывая неопределенность математического описания авиационного двигателя с известным математическим описанием регулирующего органа и датчиков возможно применение нечеткого нейронного управления на

примере контура регулирования температуры газа за турбиной низкого давления, структурная схема которого приведена на рис. 2-16.

Контур содержит регулятор температуры, селектор, дозатор, ТРДД, датчик температуры, который не показан. Регулятор температуры содержит адаптивный нейрон и блок активации, где в качестве активационных функций принят гиперболический тангенс.

Адаптивный нейрон содержит фаззификатор, синапсы по числу проекций входного вектора на выходе фаззификатора, сумматор, элемент сравнения, блок адаптации для настройки синапсов. Блок адаптации реализует алгоритм с последовательным обучением.

Данный метод относится к методам последовательной идентификации линейных систем в реальном времени. Метод последовательного обучения отличается от метода стохастической аппроксимации характеристиками сходимости. Этот метод удобно применять для процессов с медленно изменяющимися параметрами.

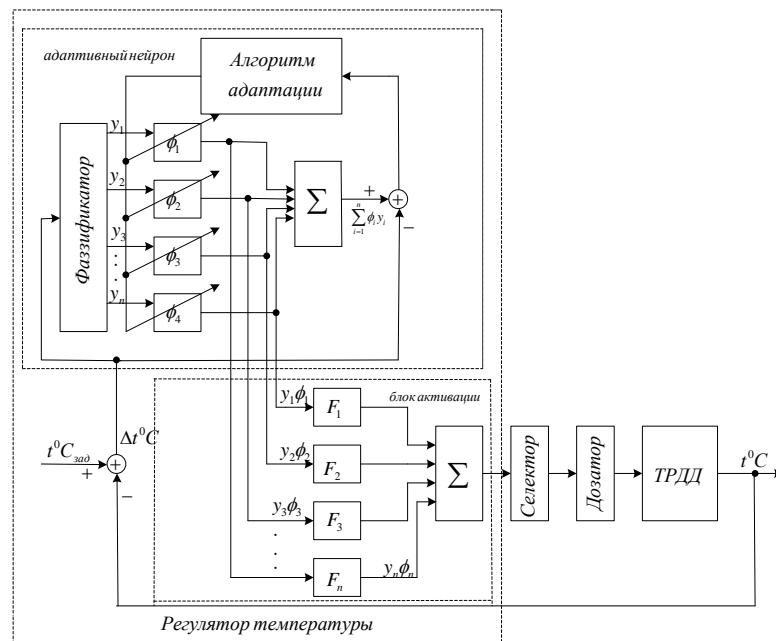


Рис. 2-16. Структурная схема контура температуры газа

Задача блока адаптации состоит в изменении коэффициентов (синапсов) нейрона ϕ_i с целью отработки отклонения между выходом сумматора нейрона и сигналом ΔT_T , представляющим разность сигналов задания и выхода объекта. Алгоритм адаптации подстраивает коэффициенты

ϕ_i так, чтобы свести ошибку $e(k) = \Delta T_T - \sum \phi_i y_i$ до некоторой заданной малой величины.

Терм-множество фаззификатора адаптивного нейрона показано на рис.2-17, а. Фаззификация выполнена на синглетонной базе, где синглетон синхронно перемещается с входным сигналом.

На рис. 2-18 приведены осциллограммы переходного процесса в контуре температуры газа с терм-множеством фаззификатора регулятора (см. рис.2-17), где показано изменения температуры газа за турбиной низкого давления при изменении $\alpha_{p,в} = 0 - 53 \text{ град}$.

Осциллограмма подтверждают работоспособность разработанного адаптивного регулятора температуры газа последовательного обучения, синапсы по числу проекций входного вектора, терм-множество фаззификатора (см. рис. 2-18, б), дозатор с обратной связью и датчиком частоты вращения вентилятора (на схеме не показан) и дефаззификатор.

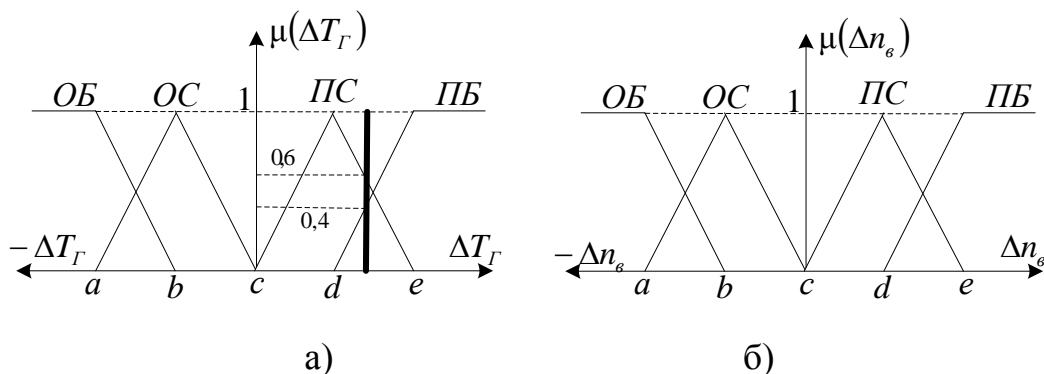


Рис. 2-17. Терм-множество фаззификатора:

а) терм-множество фаззификатора по температуре газа;

б) терм-множество фаззификатора по частоте вращения ротора вентилятора

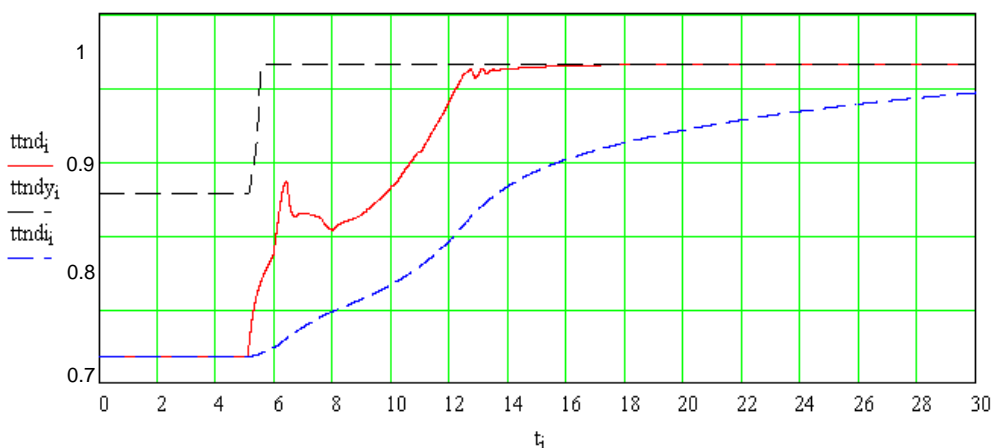


Рис. 2-18. Осциллограмма фактического изменения температуры газа за турбиной низкого давления:

$ttnd$ – текущее значение изменение температуры газа;

$ttndy$ – установочное значение температуры газа

Построение адаптивных систем управления возможно применением отдельного типа нечетких нейронных продукционных сетей типа *Anfis*, обучающихся на основе градиентных алгоритмов [5].

2.7. Нечеткая нейронная продукционная сеть *Anfis*

Нечеткая нейронная продукционная сеть *Anfis* предназначена для реализации нечеткого регулятора с предварительным обучением, чтобы придать свойства адаптации нечеткому регулятору [5, 6]. При этом возможны модификации сети *Anfis* в зависимости от применения алгоритмов нечеткого вывода: Сугено-Такаги, Сугено-Такаги-Канга (*TSK*) и Ванга-Менделя.

2.7.1. Нечеткая нейронная продукционная сеть *Anfis*

с применением алгоритма Сугено-Такаги

Рассмотрим вариант реализации сети *Anfis* с применением алгоритма нечеткого вывода Сугено-Такаги.

В модели вывода Сугено-Такаги используется набор правил:

$$R_i: \text{ЕСЛИ } x_1 \text{ это } A_{i1}, \dots \text{И } x_n \text{ это } A_{in}, \text{ ТО } y = f(X),$$

где $X = (x_1, x_2, \dots, x_n)$;

$f(X)$ – некоторая четкая функция, например, полином первого порядка.

Определяются уровни «отсечения» a_i для левой части каждого из правил согласно выражению $a_i = \min_j(A_{ij}(x_j))$, $i=1, \dots, m$, $j=1, \dots, n$ и рассчитываются «индивидуальные» выходы правил R_i , $i=1, \dots, m$:

$$y_i^* = p_{i0} + \sum_{j=1}^n p_{ij}x_j,$$

где p_{i0} , p_{ij} – коэффициенты полинома или цифровые веса, которые уточняются в процессе анализа данных.

Блок дефаззификации осуществляет переход от нечеткого значения лингвистической переменной (управление) к числовому значению. В случае упрощенного алгоритма нечеткого вывода (алгоритм Сугено нулевого порядка), когда $y_i = f(X) = p_{i0}$, $i=1, \dots, m$ следует

$$y(x_1, x_2, \dots, x_n) = \frac{\sum_{i=1}^m \min_j(\mu_{ij}(x_j)) p_{i0}}{\sum_{i=1}^m \min_j(\mu_{ij}(x_j))}$$

На рис. 2-19 изображена сеть *ANFIS* с двумя входными лингвистическими переменными x_1, x_2 , и четырьмя нечеткими правилами.

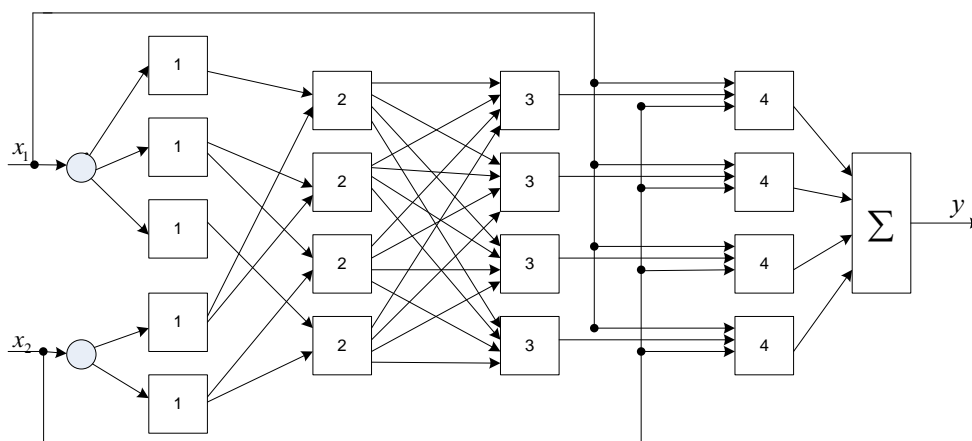


Рис. 2-19. Нечеткая нейронная сеть *ANFIS* с применением алгоритма Сугено-Такаги нулевого порядка: 1 – функции Гаусса; 2 – antecedentes; 3 – нормализация сигналов; 4 - заключения правил

Сеть *ANFIS* есть пятислойная искусственная НС прямого распространения сигнала, алгоритм реализации следующий:

- первый слой – термы входных переменных;

- второй слой – посылки (антецеденты) нечетких правил;
- третий слой нормализация степеней выполнения правил;
- четвертый слой – заключения правил;
- пятый слой – агрегирование (композиция) результата, полученного по различным правилам.

Слой 1. Входы сети соединены только со своими термами. Количество узлов первого слоя равно сумме мощностей терм-множеств входных переменных, где операция фаззификации выполнена на синглетонной базе.

Слой 2. Количество узлов второго слоя m . Каждый узел этого слоя соответствует одному нечеткому предиктивному правилу.

Узел второго слоя соединен с теми узлами первого слоя, которые формирует посылки соответствующего правила. Следовательно, каждый узел второго слоя может принимать от 1 до n сигналов. Выходом узла является степень выполнения правила, которая рассчитывается как произведение входных сигналов (по Ларсену). Обозначим выходы узлов этого слоя τ_r , $r=1, \dots, \bar{m}$, где \bar{m} – количество нечетких правил.

Слой 3. Количество узлов третьего слоя равно \bar{m} . Каждый узел этого слоя рассчитывает относительную степень выполнения нечеткого правила (нормализация) по формуле

$$\tau_r^* = \frac{\tau_r}{\sum_{j=1}^{\bar{m}} \tau_j}.$$

Слой 4. Количество узлов слоя также равно m . Каждый узел соединен с одним из узлов третьего слоя, а также со всеми входами сети. Узел четвертого слоя рассчитывает вклад одного нечеткого правила в выход сети по формуле

$$y_r = \tau_r^* (b_{0,r} + b_{1,r}x_1 + \dots + b_{n,r}x_n).$$

Слой 5. Единственный узел этого слоя суммирует вклады всех правил.

$$y = \sum_{j=1}^m y_j.$$

Настройка сети *ANFIS* с двумя входными лингвистическими переменными x_1, x_2 и четырьмя нечеткими правилами выполняется комбинацией градиентного спуска в виде алгоритмов ОРО и МНК.

Алгоритм ОРО настраивает параметры антецедентов (предпосылок), т.е. функций принадлежности фаззификатора. МНК оценивает коэффициенты заключения правил, так как они линейно связаны с выходом сети. Каждая итерация процедуры настройки выполняется в два этапа.

На **первом** этапе на входы подается обучающая выборка и по невязке между желаемым и действительным поведением сети МНК находятся оптимальные параметры узлов четвертого слоя. На **втором** этапе остаточная невязка передается с выхода сети на входы и методом ОРО модифицируются параметры узлов первого слоя. При этом найденные на предыдущем этапе коэффициенты заключения правил не изменяются. Итерационная процедура настройки продолжается, пока невязка превышает заранее установленное значение. Для настройки функций принадлежности фаззификатора, кроме метода ОРО, могут использоваться и другие алгоритмы оптимизации (пакет *NCD-Blockset*).

Рассмотрим пример применения сети *ANFIS*, где использован алгоритм Сугено-Такаги.

Пример 2–6. Разработать нечеткий нейронный регулятор частоты вращения силовой турбины турбовального двигателя (ТВД) в среде *MATLAB-SIMULINK*, где блок нечеткого вывода реализован на базе сети *ANFIS* с применением алгоритма Сугено-Такаги.

Контур управления частотой вращения силовой турбины является основным на установившихся режимах работы ТВД с несущим винтом (НВ), а также характеризуется наличием значительных возмущающих воздействий со стороны изменения угла установки лопастей НВ. Математическая модель ТВД с НВ описывается системой нелинейных дифференциальных уравнений с переменными коэффициентами. Нагрузкой для ТВД является мощность НВ, которая определяется изменением угла поворота рабочих лопастей НВ:

$$N_{\text{НВ}} = f(\varphi, n_{\text{ст}}, T_{\text{Н}}, P_{\text{Н}}),$$

где $N_{\text{НВ}}$ – мощность НВ (л.с.);

φ – угол поворота рабочих лопастей НВ (град);

$n_{\text{ст}}$ – частота вращения силовой турбины (%);

$T_{\text{Н}}$ и $P_{\text{Н}}$ – соответственно температура (K) и давление воздуха (кгс/см²) на высоте полета h ;

f – нелинейная функция.

Взаимодействие силовой турбины и НВ выражается уравнением динамики:

$$Jn_{\text{ст}} \frac{dn_{\text{ст}}}{dt} = N_e - N_{\text{НВ}},$$

где J – суммарный приведенный к валу силовой турбины момент инерции (кгс·м);

N_e – мощность силовой турбины (л.с.).

В существующей системе автоматического управления (САУ) контур регулирования частоты вращения силовой турбины реализован на базе одноконтурной системы автоматического регулирования с отрицательной обратной связью и устройством ввода воздействия (УВВ) парирующее влияние изменения угла НВ на частоту вращения силовой турбины ТВД.

Как показали результаты математического моделирования, из-за нелинейности характеристик объекта управления применение линейных законов управления не позволяет достигать заданной точности управления. Переходные процессы в существующем контуре имеют забросы и провалы с аperiodическим затуханием порядка 10 с. Известно, что нелинейные законы управления во многих случаях обеспечивают лучшее качество регулирования. С целью снижения забросов и провалов частоты вращения силовой турбины ТВД а также времени регулирования предлагается применение нечеткого нейронного регулятора. В данном регуляторе частоты вращения использована сеть *ANFIS (Adaptive Network-based Fuzzy Inference System)* с применением алгоритм нечеткого вывода Сугено-Такаги, которая

входит в состав пакета *Fuzzy Logic Toolbox* среды *Matlab-Simulink* [6]. Причины, по которой использован алгоритм Сугено-Такаги, являются следующие: возможность обучения с применением метода обратного распространения ошибки в качестве алгоритма настройки; неопределенность терм-множества дефаззификатора (дозатора).

Для оптимизации нечеткого нейронного регулятора использован инструментальный пакет *Nonlinear Control Design Blockset (NCD-Blockset)* среды *Matlab-Simulink*, который предоставляет пользователю графический интерфейс для настройки параметров динамических объектов, обеспечивающих желаемое качество переходных процессов. В качестве средства для достижения указанной цели принимается оптимизационный подход, обеспечивающий минимизацию функции штрафа за нарушение динамических ограничений. При помощи данного инструмента можно настраивать параметры нелинейной *Simulink*-модели, в качестве которых может быть задано любое количество переменных. Задание динамических ограничений осуществляется в визуальном режиме. На базе этих ограничений *NCD-Blockset* автоматически генерирует задачу конечномерной оптимизации так, чтобы точка экстремума в пространстве настраиваемых параметров соответствовала выполнению всех требований, предъявляемых к качеству процесса. Эта задача решается с привлечением специализированной процедуры квадратичного программирования из пакета *Optimization Toolbox*. Ход оптимизации контролируется на экране с помощью отображения графика контролируемого процесса и текущих значений минимизируемой функции. По завершении процесса его результат фиксируется в рабочем пространстве.

На рис. 2-20 показана схема включения нечеткого нейронного регулятора в контур управления частотой вращения силовой турбины ТВД.

Рассмотрим систему регулирования, состоящую из контура ограничения ускорения частоты вращения турбокомпрессора dn_{mk}/dt

(классический ПИ-регулятор) и контура частоты вращения силовой турбины (нечеткий нейронный регулятор).

Структура нечеткого нейронного регулятора включает в себя сеть *ANFIS*, реализующую алгоритм нечеткого вывода Сугено-Такаги нулевого порядка, показанную на рис.2-21.

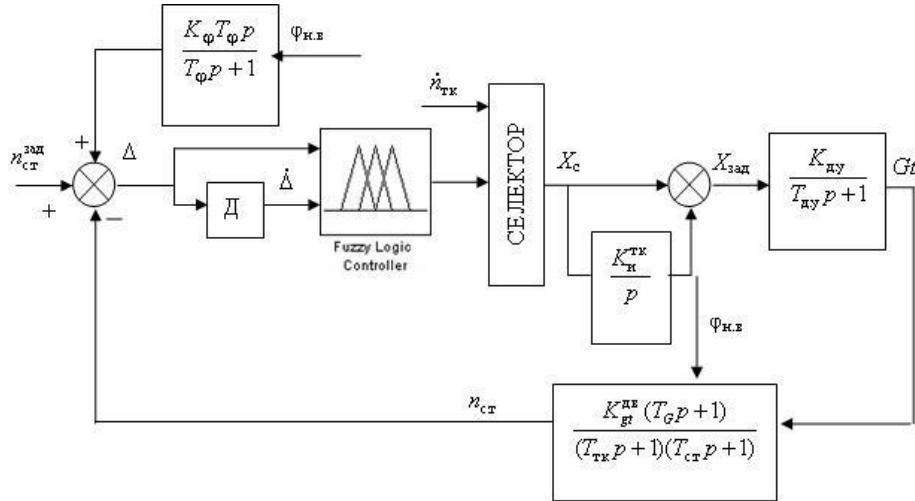


Рис.2-20. Структурная схема контура частоты вращения силовой турбины с нечетким нейронным регулятором

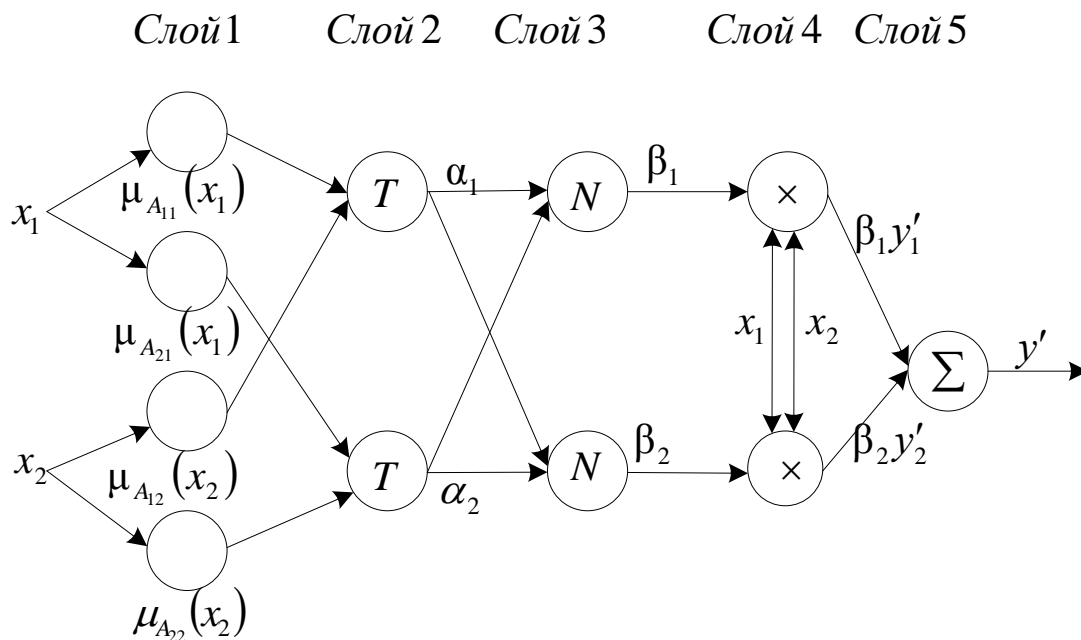


Рис. 2-21. Структура сети *ANFIS* с применением алгоритма Сугено-Такаги нулевого порядка: Т - агрегирование; N- нормализатор

База знаний такой системы содержит два нечетких правила ЕСЛИ-ТО:

П1: если x_1 есть A_{11} И x_2 есть A_{12} ТО $y_1 = a_1x_1 + b_1x_2$;

П2: если x_1 есть A_{21} И x_2 есть A_{22} ТО $y_2 = a_2x_1 + b_2x_2$.

1-й слой: Каждый узел данного слоя является настраиваемым узлом с «колоколообразной» функцией принадлежности.

2-й слой: Каждый узел данного слоя является фиксированным узлом, перемножающим входные сигналы (по Ларсену), причем выходное значение узла представляет собой вес некоторого правила:

$$\alpha_1 = A_{11}(x_1) \cdot A_{12}(x_2),$$

$$\alpha_2 = A_{21}(x_1) \cdot A_{22}(x_2)$$

3-й слой: Элементы данного слоя вычисляют нормализованные значения β_i , чтобы сумма которых равнялась единице:

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2}, \quad \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2}$$

4 и 5-й слой: Элементы этих слоев позволяют сформировать на выходе сети дефазифицированное значение:

$$y' = \frac{\alpha_1 y_1 + \alpha_2 y_2}{\alpha_1 + \alpha_2} = \frac{\alpha_1 (a_1 x_1 + b_1 x_2) + \alpha_2 (a_2 x_1 + b_2 x_2)}{\alpha_1 + \alpha_2} = \beta_1 y'_1 + \beta_2 y'_2$$

Для проверки работоспособности нечеткого нейронного регулятора и сравнения результатов моделирования с классическим ПИ-регулятором в *Matlab-Simulink* составлена математическая модель исследуемой системы автоматического регулирования, которая включает нелинейную модель ТВД, нелинейную модель дозатора, модель электронного регулятора двигателя ТВД, включающая контур ускорения частоты вращения турбокомпрессора, и контур частоты вращения силовой турбины ТВД. Модели двигателя и гидромеханической части (дозатора) рассчитываются с шагом интегрирования 0,004 с, модель электронного регулятора рассчитывается с шагом 0,02 с. Настройка нечеткого нейронного регулятора осуществляется по имеющимся экспериментальным данным работы классического ПИ-

регулятора заключается в определении количества правил алгоритмом структурной идентификации, коэффициентов линейных уравнений методом наименьших квадратов, параметров функций принадлежности методом ОРО с помощью *ANFIS*-редактора.

Для получения информативного множества экспериментальных данных на вход модели подаются входные тестирующие сигналы. Входной сигнал «изменение угла несущего винта» представляет собой периодический прямоугольный сигнал со случайной амплитудой в диапазоне от 0 град до 20 град. Время нарастания фронта 3 с, время спада фронта 2 с. Входные сигналы «температура и давление воздуха на входе в двигатель» соответствуют изменению высоты полета от 0 км до 4 км. Используется 9000 точек экспериментальных данных. Экспериментальные данные разделяются на обучающую и тестовую выборки. В обучающую выборку входит первая половина данных, а в тестовую – вторая. Перед обучением входные и выходные данные нечеткой нейронной сети нормируются в единичный интервал.

В результате работы *ANFIS*-редактора была получена сеть, реализующая алгоритм нечеткого вывода Сугено-Такаги первого порядка. Сеть имеет две входные переменные: ошибка, представляющая собой рассогласование между требуемой и фактической частотой вращения силовой турбины; скорость изменения ошибки. Терм-множество лингвистических входных переменных содержит по два термина с гауссовыми функциями принадлежности. База знаний содержит четыре предиктивных правила.

Оптимизация выполнялась для земных условий при $T_H = 288 \text{ K}$, $P_H = 1,033 \text{ кгс/см}^2$ при возмущении по углу НВ:

- на приемистости, φ : с 0 град. на 16 град. за 3 с.:
- на сбросе, φ : с 16 град. на 0 град за 2 с.

Заданное (программное) значение частоты вращения силовой турбины – 100 %. Моменты подачи возмущений по углу НВ изменяющих потребляемую мощность НВ – 15 с. и 40 с.

С помощью пакета *NCD-Blockset* полученная сеть была оптимизирована в замкнутой системе регулирования. Всего в процессе оптимизации настраивалось двенадцать переменных: восемь параметров функций принадлежности входных переменных (по два параметра на каждую из четырех функций) и четыре константы в функциях принадлежности выходной переменной.

На рис. 2-22 представлены переходные процессы в начале и в конце процесса оптимизации нечеткого нейронного регулятора в *NCD-Blockset* в системе регулирования с УВВ.

На рис. 2-23 представлены переходные процессы в системе регулирования с классическим ПИ-регулятором, с ненастроенным и настроенным нечетким нейронным регулятором, где видно, что время регулирования для нечеткого нейронного регулятора, синтезированного по данным работы классического ПИ-регулятора, оказалось меньше, а забросы и провалы частоты вращения силовой турбины ТВД больше, чем у классического ПИ-регулятора. После оптимизации нечеткого нейронного регулятора в *NCD-Blockset* время регулирования, а также забросы и провалы частоты вращения силовой турбины ТВД удовлетворяют заданным в *NCD-Blockset* динамическим ограничениям.



Рис. 2-22. Процесс оптимизации нечеткого нейронного регулятора в *NCD-Blockset* в системе регулирования с УВВ

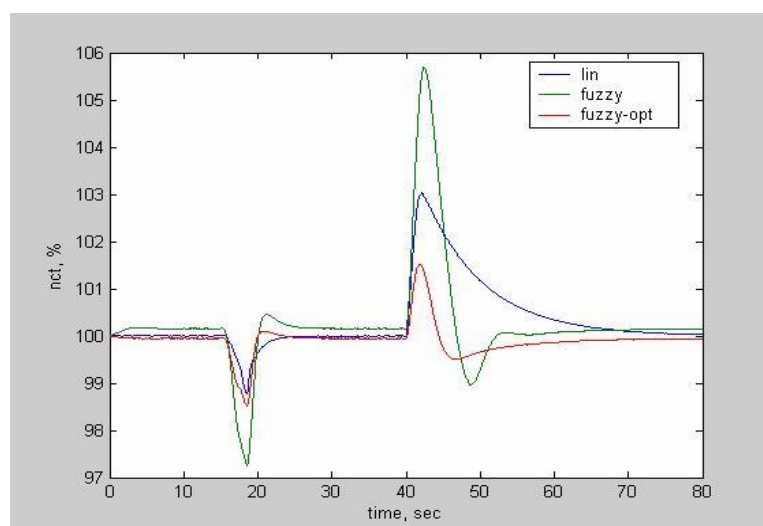


Рис. 2-23. Переходные процессы в системе регулирования с классическим ПИ-регулятором и с ненастроенным и настроенным нечетким нейронным регулятором

Также была исследована работа нечеткого нейронного регулятора в системе регулирования без учета измерителя угла НВ. В качестве нечеткого нейронного регулятора использовалась сеть, полученная в результате работы *ANFIS*-редактора при синтезе регулятора с УВВ. В процессе оптимизации настраивалось так же двенадцать переменных.

Также была исследована работа нечеткого нейронного регулятора в системе регулирования без учета измерителя угла НВ. В качестве нечеткого нейронного регулятора использовалась сеть, полученная в результате работы

ANFIS-редактора при синтезе регулятора с УВВ. В процессе оптимизации настраивалось так же двенадцать переменных.

На рис. 2-24 представлены переходные процессы в начале и в конце процесса оптимизации нечеткого нейронного регулятора в *NCD-Blockset* в системе регулирования без УВВ.

На рис. 2-25 представлены переходные процессы в системе регулирования с классическим ПИ-регулятором без УВВ и с настроенным нечетким нейронным регулятором без УВВ. После оптимизации нечеткого нейронного регулятора в пакете *NCD-Blockset* время регулирования, а также забросы и провалы частоты вращения силовой турбины удовлетворяют заданным в *NCD-Blockset* ограничениям.

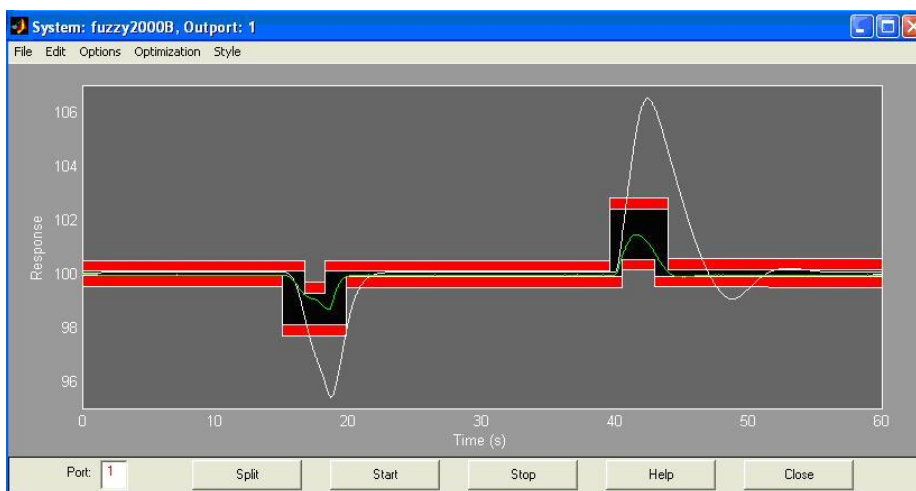


Рис. 2-24. Процесс оптимизации нечеткого нейронного регулятора в *NCD-Blockset* в системе регулирования без УВВ

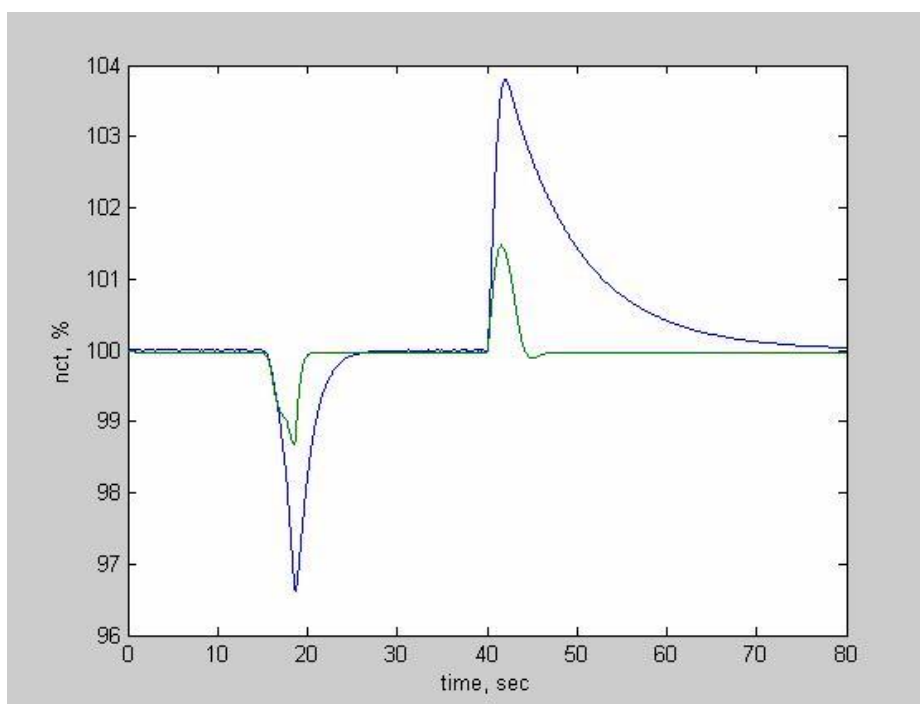


Рис.2-25. Переходные процессы в системе регулирования с классическим регулятором без УВВ и с настроенным нечетким нейронным регулятором без УВВ

Таким образом, применение сети *ANFIS* в качестве регулятора в системе регулирования без УВВ позволяет получить сравнимые, с системой регулирования с УВВ, показатели качества переходных процессов.

Результаты моделирования показали, что разработанный нечеткий нейронный регулятор частоты вращения вентилятора авиационного двигателя обеспечивают заданное качество переходных процессов при возмущениях по углу НВ, отличных от заданных, при выполнении оптимизации.

Рассмотрим сеть *Anfis* с применением алгоритма Такаги-Сугено-Канга, который реализован нечеткой нейронной сетью, где процесс обучения разбит на два этапа и процесс вычислений по этапам выполняется параллельно и одновременно.

2.7.2. Нечеткая нейронная продукционная сеть *Anfis* с применением алгоритма Такаги-Сугено-Канга

*Описание нечеткой нейронной продукционной сети *Anfis* с применением*

алгоритма Такаги-Сугено-Канга.

Отличие алгоритмов Такаги-Сугено-Канга (TSK) [5] от алгоритма Сугено-Такаги заключается в реализации нечеткой продукционной модели, основанной на правилах типа

P_i : Если x_1 есть A_{i1} И...И x_j есть A_{ij} И...И x_m есть A_{im} , ТО

$$y = c_{i0} + \sum_{j=1}^m c_{ij} x_j, j = 1, \dots, n.$$

Данная нечеткая адаптивная сеть базируется на следующих положениях:

- входные переменные являются четкими;
- функции принадлежности всех перечисленных множеств определены функцией Гаусса;

$$\mu_{A_{ij}}(x_j) = \exp\left(-0,5\left(\frac{x_j - a_{ij}}{b_{ij}}\right)^2\right),$$

где x_j – входы сети;

a_{ij}, b_{ij} – настраиваемые параметры функции принадлежности.

- нечеткая импликация Ларсена – нечеткое произведение;
- T -норма – нечеткое произведение;
- композиция не производится;
- метод дефаззификации – метод центроида.

Исходя из этих положений функциональная зависимость для получения выходной переменной величины после дефаззификации примет вид [5]

$$y' = \frac{\sum_i^n \left(\left(c_{i0} + \sum_{j=1}^m c_{ij} x_j \right) \prod_j^m \mu_{A_{ij}}(x'_j) \right)}{\sum_{i=1}^n \prod_j^m \mu_{A_{ij}}(x'_j)} =$$

$$= \frac{\sum_i^n \left(\left(c_{i0} + \sum_{j=1}^m c_{ij} x_j \right) \prod_{j=1}^m \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right] \right)}{\sum_{i=1}^n \prod_j^m \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}. \quad (2-11)$$

Структура нечеткой нейронной продукционной сети Anfis с применением алгоритма TSK

На рис. 2-26 представлена сеть *Anfis* с применением алгоритма *TSK*.

Данное аналитическое выражение (2-11) лежит в основе сети *Anfis* с применением алгоритма *TSK*, которая включает в себя пять слоев.

Слой 1 состоит из элементов, которые выполняют фаззификацию входных четких переменных x'_j ($j=1, \dots, n$). Элементы этого слоя вычисляют значения степеней принадлежности функций принадлежности $\mu_{A_j}[x'_j]$, заданных гауссовскими функциями с параметрами a_{ij} и b_{ij} .

Слой 2, число элементов которого равно количеству правил в базе, выполняет нечеткую импликацию степеней принадлежности соответствующих правил.

Слой 3 генерирует значения функций $\left(c_{j0} + \sum_{j=1}^m c_{ij} x'_j \right)$, которые умножаются на результаты вычислений элементами предыдущего слоя.

В **слое 4** первый элемент (сумматор) служит для активизации заключений правил в соответствии со значениями агрегированных в предыдущем слое степеней принадлежности предпосылок правил. Вторым элементом (сумматор) проводит вспомогательные вычисления для последующей дефаззификации результата.

Слой 5 состоит из одного нормализующего элемента и выполняет дефаззификацию результата.

Из приведенного описания следует, что сеть *TSK* содержит два параметрических слоя (слой 1 и 3). Настраиваемыми в процессе обучения параметрами являются:

– в слое 1 – нелинейные параметры a_{ij} и b_{ij} гауссовских функций принадлежности фаззификатора;

– в слое 3 – параметры c_{i0} и c_{ij} линейных функций $\left(c_{j0} + \sum_{j=1}^m c_{ij}x'_j \right)$ из заключений правил.

При наличии n правил и m входных переменных число параметров первого слоя равно $2nm$, а второго – $n(m+1)$. Таким образом, суммарное число настраиваемых параметров равно $n(3m+1)$.

Сначала рассчитываются параметры c_{i0} и c_{ij} линейных функций при условии фиксированных значений параметров a_{ij} и b_{ij} . Параметры c_{i0} и c_{ij} находятся путем решения системы линейных уравнений.

Представим выходную переменную из выражения (2-12) в следующем виде

$$y' = \sum_{i=1}^n w'_i \left(c_{i0} + \sum_{j=1}^m c_{ij}x'_j \right), \quad (2-12)$$

где

$$w'_i = \frac{\prod_{j=1}^m \mu_{A_{ij}}(x'_j)}{\sum_{i=1}^n \prod_j \mu_{A_{ij}}(x'_j)} = \frac{\prod_j \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}{\sum_{i=1}^n \prod_j \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]} = \text{const}$$

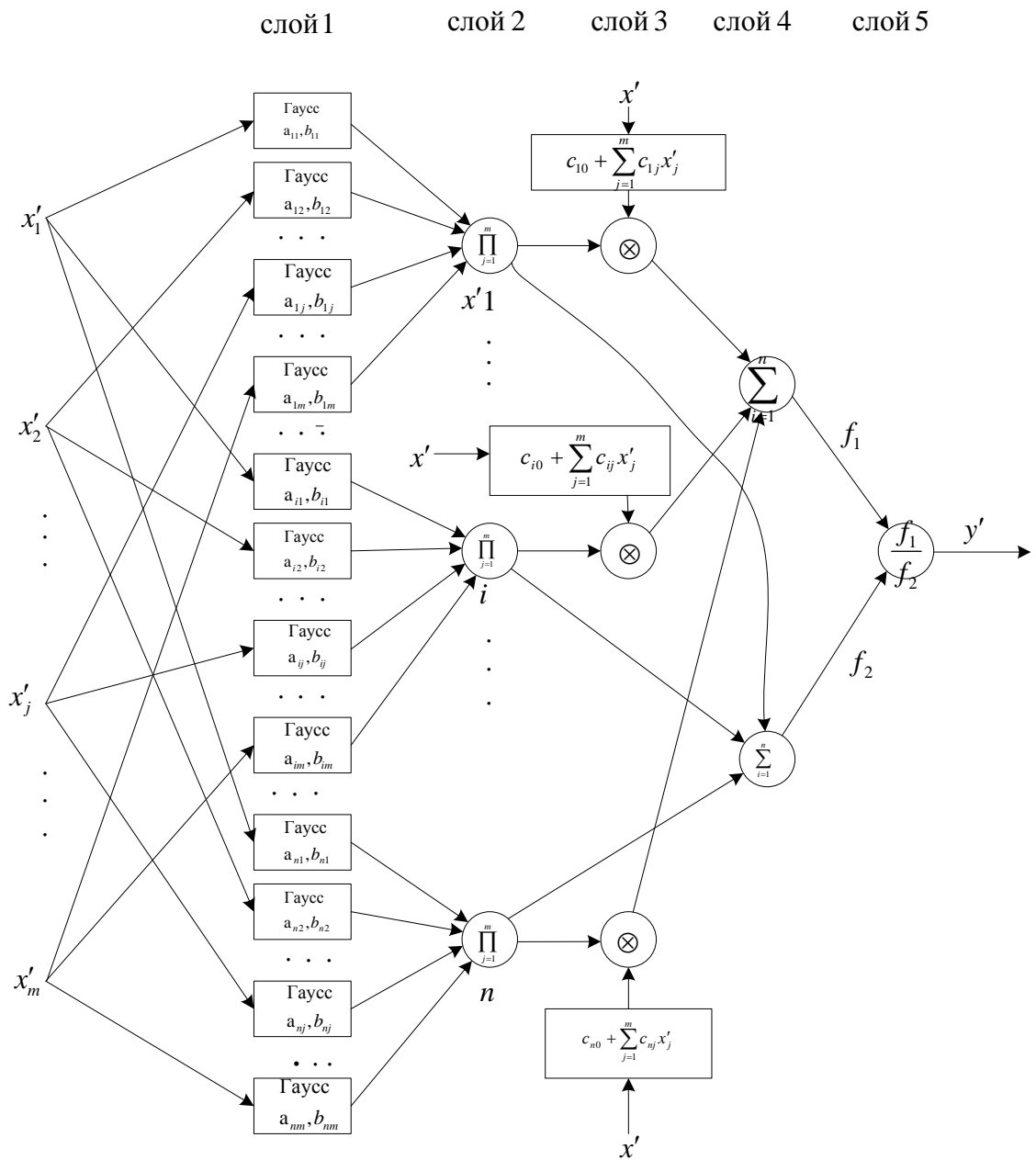


Рис. 2-26. Структура нечеткой нейронной продукционной сети *Anfis* с применением алгоритма *TSK*

Алгоритм обучения сети Anfis с применением алгоритма TSK.

При K обучающих примерах $(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}, y^{(k)})$, где $k=1, \dots, K$, и замене значений выходных переменных $y'^{(k)}$ значениями эталонных переменных $y^{(k)}$, получим систему из K линейных уравнений вида (2-13):

$$\begin{bmatrix} w_1^{(1)} & w_1^{(1)} x_1^{(1)} & \dots & w_1^{(1)} x_m^{(1)} & \dots & w_n^{(1)} & w_n^{(1)} x_1^{(1)} & \dots & w_n^{(1)} x_m^{(1)} \\ w_1^{(2)} & w_1^{(2)} x_1^{(2)} & \dots & w_1^{(2)} x_m^{(2)} & \dots & w_n^{(2)} & w_n^{(2)} x_1^{(2)} & \dots & w_n^{(2)} x_m^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ w_1^{(k)} & w_1^{(k)} x_1^{(k)} & \dots & w_1^{(k)} x_m^{(k)} & \dots & w_n^{(k)} & w_n^{(k)} x_1^{(k)} & \dots & w_n^{(k)} x_m^{(k)} \end{bmatrix}' \times \begin{bmatrix} c_{10} \\ \dots \\ c_{1m} \\ \dots \\ c_{n0} \\ \dots \\ c_{nm} \end{bmatrix} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(k)} \end{bmatrix}, \quad (2-13)$$

где $w_i^{(k)}$ – агрегированная степень истинности предпосылок по i -му правилу при предъявлении k -го входного вектора $(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)})$.

Запишем (2-13) в сокращенном матричном виде

$$W \times c = y.$$

Размерность матрицы W равна $K \times (m+1)n$ при этом обычно количество строк K значительно больше количества столбцов: $K \times (m+1)n$. Решение этой системы уравнений можно провести за один шаг при помощи псевдоинверсии матрицы W :

$$c = W^+ y = (W^T \cdot W)^{-1} W^T y.$$

Затем после определения линейных параметров c_{ij} их фиксируют и рассчитывают фактические выходные сигналы сети для всех примеров, для чего используется линейная зависимость

$$y' = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(k)} \end{bmatrix} = W \cdot c.$$

Определяем вектор ошибок:

$$e = y' - y.$$

После чего, например, по алгоритму Уидроу-Хоффа уточняем параметры:

$$a_{ij}^{(k)}(t+1) = a_{ij}^{(k)}(t) - C \frac{dE^{(k)}(t)}{da_{ij}^{(k)}};$$

$$b_{ij}^{(k)}(t+1) = b_{ij}^{(k)}(t) - C \frac{dE^{(k)}(t)}{db_{ij}^{(k)}}.$$

После уточнения нелинейных параметров процесс адаптации параметров запускается вновь до тех пор, пока не наступит повторяемость результатов. Данный алгоритм называют гибридным. Его особенность заключается в разделении этапов процесса обучения. Гибридный алгоритм более эффективен, чем метод Уидроу-Хоффа, у которого уточнение всех параметров производится параллельно и одновременно.

В сети *TSK* результатом является полином $c_{i0} + \sum_{j=1}^m c_{ij}x_j$, тогда как в сети Ванга-Менделя выходная переменная представляется константной c_i , которую можно рассматривать как полином нулевого порядка. Поэтому рассмотрим нечеткую нейронную продукционную сеть Ванга-Менделя как частный случай сети *TSK*.

2.7.3. Нечеткая нейронная продукционная сеть *Anfis* с алгоритмом Ванга-Менделя

Описание сети Anfis с алгоритмом Ванга-Менделя

Сеть *Anfis* с применением алгоритма Ванга-Менделя основана на правилах [5]:

P_i : Если x_i есть A_{i1} И...И x_j есть A_{ij} И...И x_m есть A_{im} , ТО $y = B_{ij}$, $j = 1, \dots, n$.

Данная нечеткая адаптивная сеть базируется на следующих положениях:

- входные переменные являются четкими;
- функции принадлежности всех перечисленных множеств определены функцией Гаусса;
- нечеткая импликация Ларсена – нечеткое произведение;
- *T*-норма – нечеткое произведение;
- композиция не производится;
- метод дефаззификации – средний центр.

Исходя из этих предпосылок нечеткий вывод для данной модели имеет следующий вид:

$$\mu_{B_i'}(y) = \sup_{x \in X} \{ \mu_{A_i'}(x) T \mu_{A_i \rightarrow B_i}(x, y) \} = \sup_{x \in X} \{ \mu_{A_i'}(x) \cdot \mu_{A_i \rightarrow B_i}(x, y) \} =$$

$$= \sup_{x \in X} \{ \mu_{A_i'}(x) \mu_{A_{ij}}(x) \mu_{B_i}(y) \} = \sup_{x_1, \dots, x_m \in X} \left\{ \mu_{B_i}(y) \prod_{j=1}^m (\mu_{A_{ij}}(x_j) \mu_{A_{ij}}(x_j)) \right\}. \quad (2-14)$$

Учитывая, что входные переменные x_1, \dots, x_m являются четкими, то (2-14) принимает следующий вид

$$\mu_{B_i'}(y) = \mu_{B_i}[y] \prod_{j=1}^m \mu_{A_{ij}}(x'_j).$$

Так аккумуляирования активизированных заключений правил не проводится, а методом дефаззификации является метод среднего центра, то выходная переменная определяется:

$$y' = \frac{\sum_{i=1}^n (\arg \max_y \mu_{B_i'}(y) \mu_{B_i}(y))}{\sum_{i=1}^n \mu_{B_i'}(y)} = \frac{\sum_{i=1}^n (\arg \max_y \mu_{B_i}(y) \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}{\sum_{i=1}^n (\mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}. \quad (2-15)$$

С учетом того, что максимальное значение, которое $\mu_{B_i}(y)$ может принять в точке $\arg \max_y \mu_{B_i}(y)$ равно единице, (2-15) примет вид (2-16)

$$y' = \frac{\sum_{i=1}^n (\arg \max_y \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}{\sum_{i=1}^n \prod_{j=1}^m \mu_{A_{ij}}(x'_j)}. \quad (2-16)$$

В случае функции принадлежности всех нечетких множеств вида функцией Гаусса, то выражение (2-16) примет вид

$$y' = \frac{\sum_{i=1}^n (\arg \max_y (\exp[-\frac{y-c_i}{d_i}])) \prod_{j=1}^m \exp[-\frac{x'_j - a_{ij}}{b_{ij}}]}{\sum_{i=1}^n \prod_{j=1}^m \exp[-\frac{x'_j - a_{ij}}{b_{ij}}]} \quad (2-17)$$

где c_i, d_i – соответственно, центры и ширина гауссовский функций, представляющих функции принадлежности нечетких множеств B_i заключений правил; a_{ij}, b_{ij} – соответственно центры и ширина гауссовских функций, представляющих функции принадлежности нечетких множеств A_{ij} предпосылок правил.

В окончательном виде (2-17) преобразуется в (2-18)

$$y' = \frac{\sum_{i=1}^n c_i \prod_{j=1}^m \exp\left[-\frac{x'_j - a_{ij}}{b_{ij}}\right]}{\sum_{i=1}^n \prod_{j=1}^m \exp\left[-\frac{x'_j - a_{ij}}{b_{ij}}\right]} . \quad (2-18)$$

Структура сети Anfis с применением алгоритма Ванга-Менделя

На рис. 2-27 представлена структура нечеткой продукционной сети

Anfis с алгоритмом Ванга-Менделя, элементы слоев которой реализуют соответствующие компоненты выражения (2-18).

В слое 2, число элементов которого равно числу количества правил в базе, осуществляется агрегирование степеней принадлежности предпосылок соответствующих правил.

В слое 3 первый элемент служит для активизации заключений правил (c_i) в соответствии со значениями агрегированных в предыдущем слое степеней принадлежности предпосылок правил. Второй элемент слоя проводит вспомогательные вычисления для последующей дефаззификации результата.

Слой 4, состоящий из одного элемента, выполняет дефаззификацию выходной переменной.

Обучение продукционной сети Anfis с применением алгоритма Ванга-Менделя

Алгоритм обучения разделяется на две процедуры. Сначала настраиваются линейные параметры элементов третьего слоя c_i , а затем – параметры нелинейной функции принадлежности в элементах первого слоя a_{ij} и b_{ij} , где $i = 1, \dots, n$; $j = 1, \dots, m$.

Этап 1. Для каждого примера из обучающей выборки $(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}, y^{(k)})$, где $k = 1, \dots, K$, рассчитываются значения выходной переменной $y'^{(k)}$.

Этап 2. Вычисляется функция ошибки для всех примеров обучающей выборки:

$$E^{(k)} = 0,5(y'^{(k)} - y^{(k)})^2, k=1, \dots, K$$

Этап 3. Корректируются значения c_i для каждого i -го правила по каждому k -му примеру обучающей выборки, исходя из соотношения

$$c_i(t+1) := c_i(t) - C \frac{dE^{(k)}(t)}{dc_i(t)}, i=1, \dots, n, k=1, \dots, K;$$

Процедура корректировки значений c_i (этапы 1-3) итерационно повторяется и считается завершенной в случае, если:

– либо значение функции ошибки по каждому примеру обучающей выборки не превышает некоторого установленного порога:

$$E^{(k)} < \varepsilon, k=1, \dots, K;$$

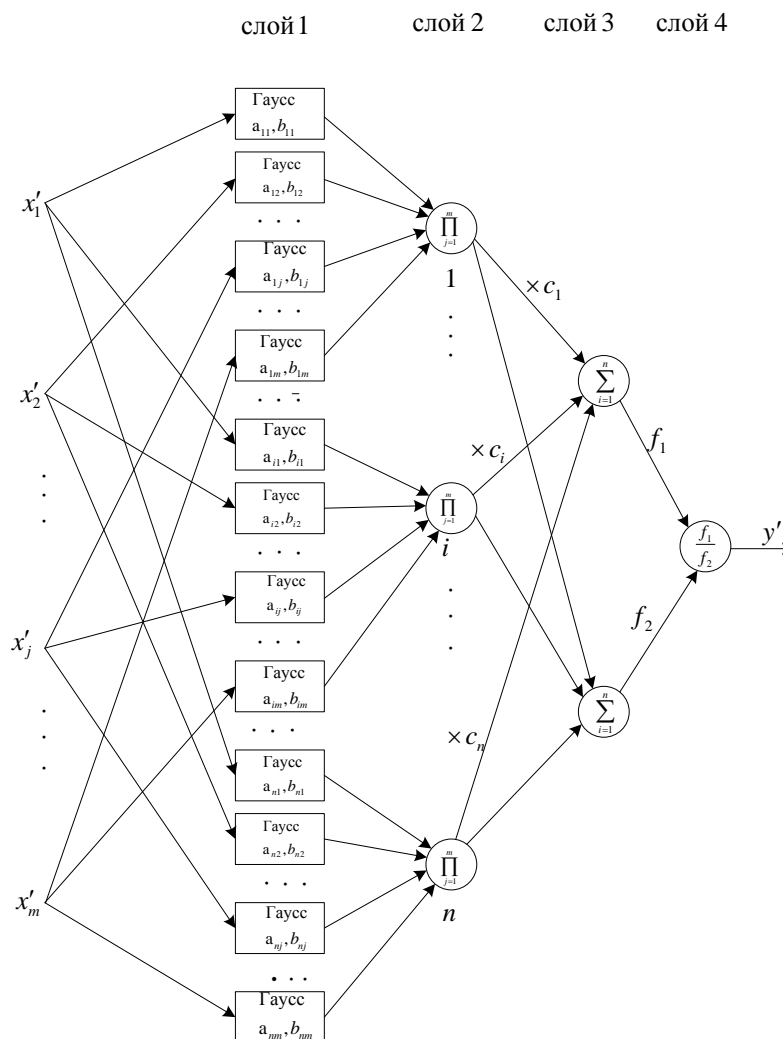


Рис.2-27. Структура нечеткой нейронной продукционной сети *Anfis* с применением алгоритма Ванга-Менделя

– либо оценка средней суммарной погрешности нечеткой продукционной модели с учетом всех примеров обучающей выборки не превышает некоторого установленного порога:

$$E = \frac{1}{K} \sum_{k=1}^K (y'^{(k)} - y^{(k)})^2 < \varepsilon ;$$

– либо погрешность стабилизировалась на некотором значении $\succ \varepsilon$.

При выполнении процедуры корректировки значений a_{ij} и b_{ij} в элементах первого слоя этапы 1 и 2 выполняются аналогично этапам процедуры корректировки c_i . На заключительном этапе этой процедуры значения a_{ij} и b_{ij} изменяются в соответствии со следующими выражениями

$$a_{ij}(t+1) := a_{ij}(t) - C \frac{dE^{(k)}(t)}{da_{ij}(t)} = a_{ij}(t) - C \frac{2(x_j'^{(k)} - a_{ij})(y'^{(k)} - y^{[k]})(c_i - y'^{(k)}) \prod_{j=1}^m \exp \left[- \left(\frac{x_j'^{(k)} - a_{ij}}{b_{ij}} \right)^2 \right]}{b_{ij}^2 \sum_{i=1}^n \prod_{j=1}^m \exp \left[- \left(\frac{x_j'^{(k)} - a_{ij}}{b_{ij}} \right)^2 \right]}$$

Условия завершения корректировки значений a_{ij} и b_{ij} подобны c_i .

В случае невыполнения первого или второго условия процесс итерационно повторяется, начиная с корректировки c_i до тех пор, пока нечеткая нейронная сеть *Anfis* не будет корректно обучена.

Сеть *Anfis* с алгоритмом Ванга-Менделя, отличаясь простотой с вычислительной точки зрения и большой чувствительностью к изменениям входных переменных, где реализован градиентный метод оптимизации фронтального типа, в то же время не является эффективным с точки зрения быстродействия.

Таким образом, на примере нечеткой нейронной сети *Anfis* и ее модификаций, показан вариант адаптации регуляторов нечеткого нейронного управления.

2.8. Модификация нечеткой продукционной сети *Anfis* и ее разновидностей с применением адаптивных нейронов

Недостаток сети *Anfis* и ее разновидностей заключается в настройке градиентными итерационными алгоритмами (ОРО, Уидроу-Хоффа), требующих значительное время на настройку и возможность управлять медленными процессами. В системах реального времени это не приемлемо, так как время на выработку управляющего воздействия жестко ограничено и задача в случае не выполнения этого требования может быть снята. Данная проблема может быть решена применением адаптивных линейных нейронов (с одним входом и нулевым смещением) [18] с алгоритмом последовательного обучения в первом слое.

Для применения алгоритма последовательного обучения необходимо:

- в первом слое модифицированной сети заменить терм-множество из нелинейных ФП (Гаусса, колоколообразные, сигмоидные и т.д.) на множество адаптивных линейных нейронов. В этом случае фаззификация в нейронной сети исключается;
- во втором слое модифицированной сети *Anfis* возможно любая нечеткая импликация (антеценденты);
- в третьем слое проводится нормализация сигналов второго слоя;
- в четвертом слое сети *Anfis* возможно применение заключений правил (например, алгоритма нечеткого вывода (Цукамото, Сугено-Такаги нулевого или первого порядка, Мамдани и т.д.).

На рис. 2-28 приведены в качестве примера структура модифицированной нечеткой нейронной продукционной сети *Anfis* с применением алгоритма *Сугено-Такаги*.

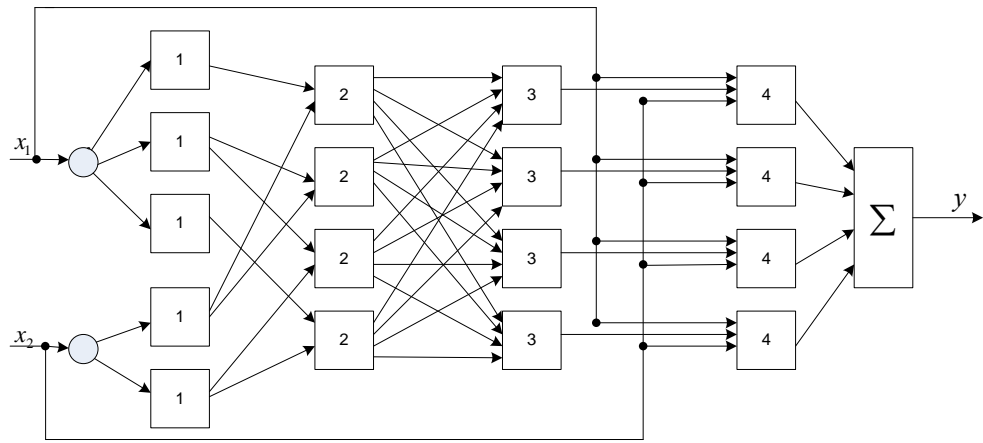


Рис. 2-28. Нечеткая нейронная сеть ANFIS с применением алгоритма Сугено-Такаги нулевого порядка: 1 – адаптивные линейные нейроны; 2 – antecedentes; 3 – нормализация сигналов; 4 - заключения правил

Рассмотрим на примере описание модифицированной сети Anfis с применением алгоритма Ванга-Менделя

Модифицированная сеть Anfis с применением алгоритма Ванга-Менделя базируется на следующих положениях:

- входные переменные являются четкими;
- определено множество адаптивных линейных нейронов;
- нечеткая импликация Ларсена – нечеткое произведение;
- T-норма – нечеткое произведение;
- композиция не производится;
- метод дефаззификации – средний центр.

Если аккумулярование активизированных заключений правил не проводится, а методом дефаззификации является метод среднего центра, то выходная переменная определяется:

$$y' = \frac{\sum_{i=1}^n (\operatorname{argmax}_y \mu_{B'_i}(y) \mu_{B'_i}(y))}{\sum_{i=1}^n \mu_{B'_i}(y)} = \frac{\sum_{i=1}^n (\operatorname{argmax}_y \mu_{B'_i}(y) \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}{\sum_{i=1}^n (\mu_{B'_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}. \quad (2-15)$$

С учетом того, что максимальное значение, которое $\mu_{B_i}(y)$ может принять в точке $\operatorname{argmax}_y \mu_{B'_i}(y)$ равно единице, (2-15) примет вид (2-16)

$$y' = \frac{\sum_{i=1}^n (\operatorname{argmax}_y \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}}(x'_j))}{\sum_{i=1}^n \prod_{j=1}^m \mu_{A_{ij}}(x'_j)} . \quad (2-16)$$

Обучение модифицированной продукционной сети Anfis с применением алгоритма Ванга-Менделя

Алгоритм обучения следующий.

Этап 1. Для каждого примера из обучающей выборки $(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}, y^{(k)})$, где $k=1, \dots, K$, рассчитываются значения выходной переменной $y'^{(k)}$. Настройка синапсов первого слоя выполняется с помощью нейронов с алгоритмами последовательного обучения за одну итерацию согласно (2.11).

На рис.2-29 приведена структурная схема адаптивного линейного нейрона. На рис. 2-30 приведена структурная схема модифицированной сети *Anfis* с применением алгоритма Ванга-Менделя.

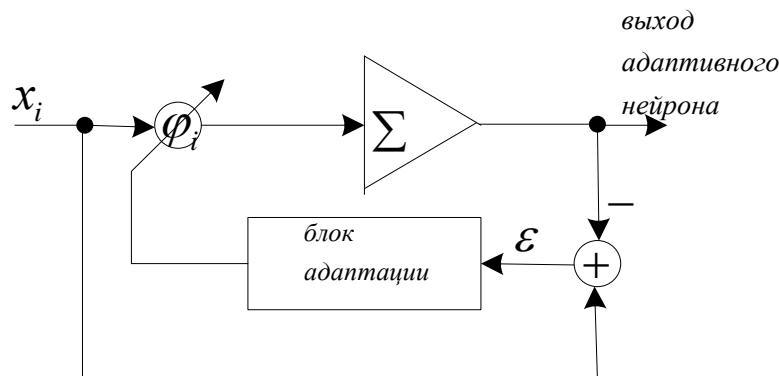


Рис. 2-29. Структурная схема адаптивного линейного нейрона

Этап 2. Вычисляется функция ошибки для всех примеров обучающей выборки:

$$E^{(k)} = 0,5(y'^{(k)} - y^{(k)})^2, \quad k=1, \dots, K$$

Этап 3. Корректируются значения c_i для каждого i -го правила по каждому k -му примеру обучающей выборки, исходя из соотношения

$$c_i(t+1) := c_i(t) - C \frac{dE^{(k)}(t)}{dc_i(t)}, \quad i=1, \dots, n, \quad k=1, \dots, K;$$

Процедура корректировки значений c_i итерационно повторяется и считается завершенной в случае, если:

– либо значение функции ошибки по каждому примеру обучающей выборки не превышает некоторого установленного порога:

$$E^{(k)} < \varepsilon, k = 1, \dots, K;$$

– либо оценка средней суммарной погрешности нечеткой продукционной модели с учетом всех примеров обучающей выборки не превышает некоторого установленного порога:

$$E = \frac{1}{K} \sum_{k=1}^K (y^{(k)} - y^{(k)})^2 < \varepsilon;$$

– либо погрешность стабилизировалась на некотором значении $> \varepsilon$.

Модифицированная сеть *Anfis* с применением алгоритма Ванга-Менделя отличается простотой и является эффективной с точки зрения быстродействия, так как время требуется только на настройку c_i .

Таким образом, на примере модифицированной нечеткой нейронной сети *Anfis* с применением алгоритма Ванга-Менделя, показано как выполнить адаптацию регуляторов нечеткого нейронного управления.

Аналогично возможна модификация нечеткой нейронной сети *Anfis* с применением алгоритма Такаги-Сугено-Канга, показанной на рис. 2-31.

Таким образом, рассмотрены основные понятия теории нечеткого нейронного управления: нейрон, методы адаптации нейронов, классификация нейронных сетей, методы обучения нейронных сетей на базе формальных нейронов, генетические алгоритмы, примеры применения нейронных сетей: адаптивные нечеткие регуляторы температуры газа авиационного двигателя, частоты вращения вентилятора авиационного двигателя. Приведенные примеры нечеткого нейронного управления применительно к авиационным двигателям показывают следующее.

1. Адаптивность управления авиационного двигателя подтверждается разработкой адаптивного регулятора частоты вращения вентилятора и адаптивного регулятора температуры газа за турбиной низкого давления.

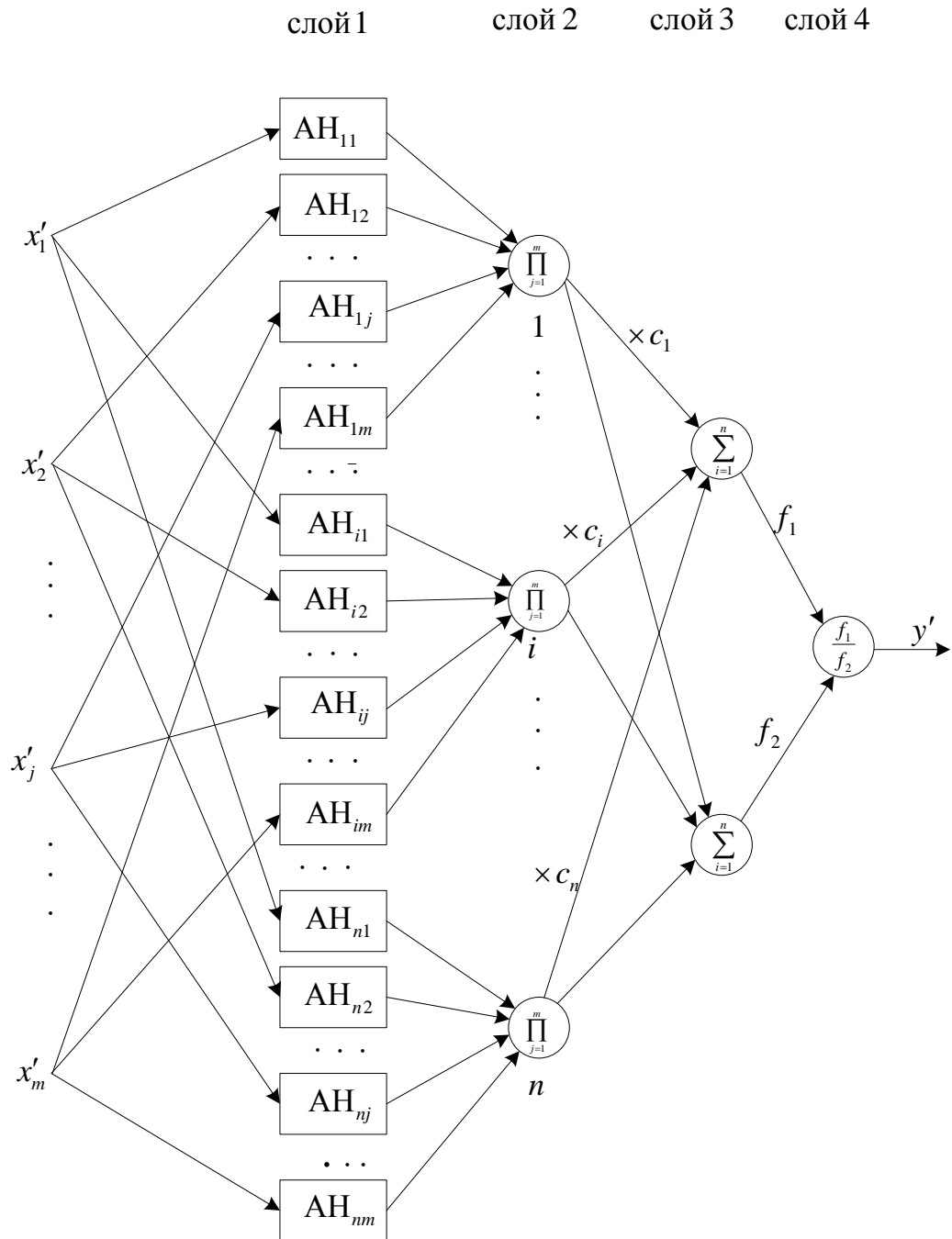


Рис.2-30. Структура модифицированной нечеткой нейронной продукционной сети ANFIS с применением алгоритма Ванга-Менделя, где АН – адаптивный линейный нейрон

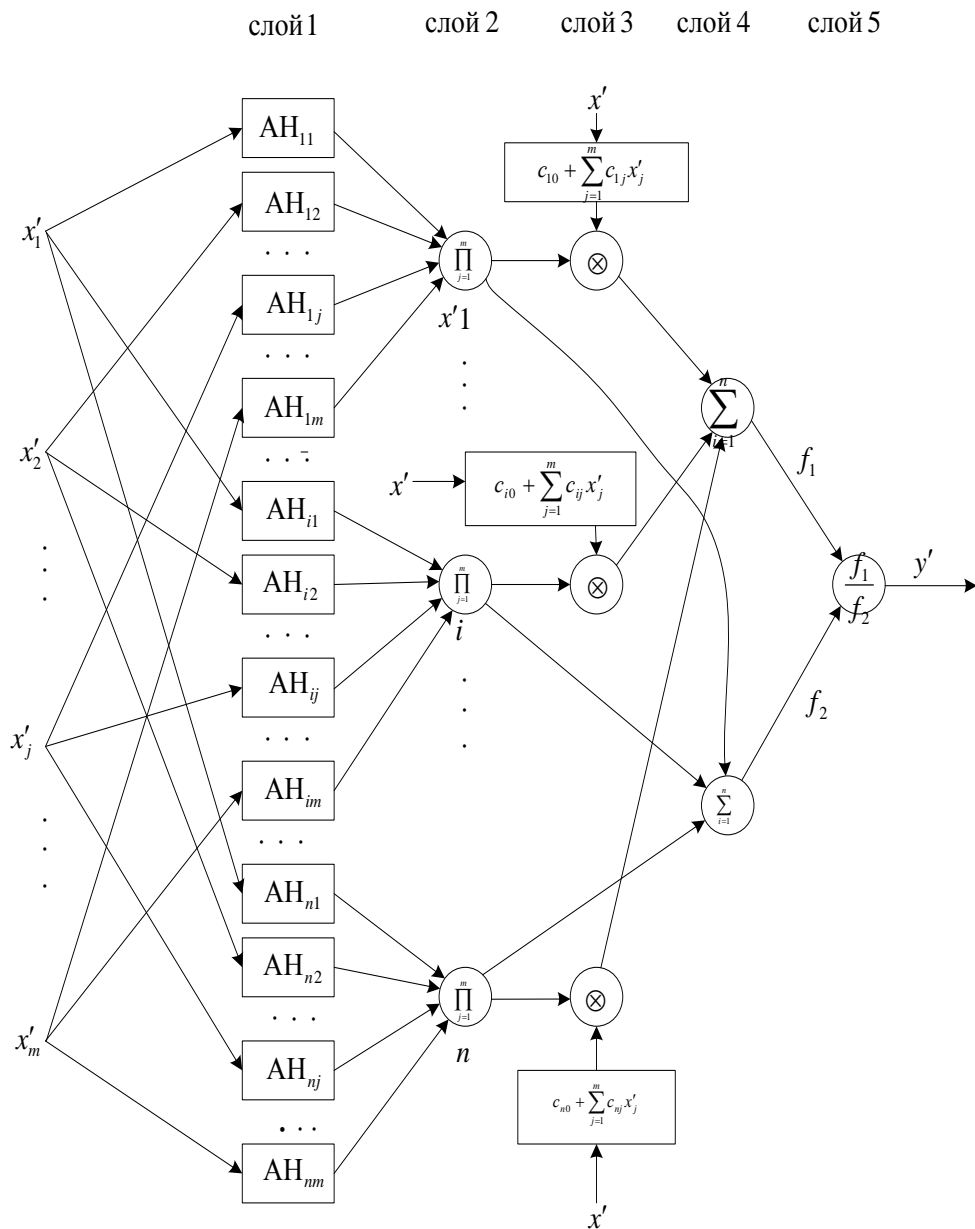


Рис. 2-31. Структура модифицированной нечеткой нейронной продукционной сети ANFIS с применением алгоритма Такаги-Сугено-Канга:

АН – адаптивный линейный нейрон; $\Pi_1 \dots \Pi_n$ – умножители;

2. Вновь спроектированный адаптивный регулятор температуры газа за компрессором низкого давления с применением нейронной технологии, обладая астатическими свойствами, исключает ошибку регулирования в статике, парирует контролируемые и неконтролируемые возмущения с максимальным быстрым действием на всех режимах работы авиационного двигателя.

3. Вновь спроектированный нейро-нечеткий регулятор частоты вращения вентилятора ТРДД с применением нейронной технологии обеспечивает заданное техническим заданием качество управления при изменении по ресурсу характеристик двигателя, как объекта управления. Регулятор парирует контролируемое и неконтролируемое изменение параметров объекта регулирования в условиях воздействия возмущающих факторов. Регулятор, обладая астатическими свойствами, исключает ошибку регулирования в статике. Переходные процессы при увеличении частоты вращения ротора вентилятора обеспечивает асимптотический запас устойчивости системы.

Сравнительный анализ нечеткого и нечетко-нейронного управления

Таблица 2-3. Различия в управлении

Нечеткое управление	Нечетко-нейронное управление
<p>1. Требуется составление предиктивных правил и формирования базы знаний, используя человеческую интуицию и опыт оператора. Альтернативой этому является формирование матрицы решений с помощью «гиперболического» закона расположения терм лингвистической переменной выхода.</p> <p>2. Обеспечивает астатизм в системе регулирования за счет статической коррекции коэффициента усиления с помощью термы НОРМА.</p> <p>3. Теряет наглядность постановки решения задачи, так как матрицы решений при трех и более лингвистических переменных является гиперобъемными.</p> <p>4. Допускает дополнительно динамическую коррекцию коэффициента усиления за счет модификации метода центроида, где поиск</p>	<p>1. Исключает составление предиктивных правил и формирования базы знаний, требующих человеческую интуицию и опыт оператора.</p> <p>2. Обеспечивает астатизм в системе регулирования за счет настройки сети градиентными алгоритмами.</p> <p>3. Имеет наглядность постановки решения задачи, так как матрица решения заменяется НС, которая располагается на плоскости, и нет ограничений на число лингвистических переменных.</p> <p>4. Допускает дополнительно динамическую коррекцию коэффициента усиления за счет коррекции активационной функции последнего нейрона НС.</p> <p>5. Является адаптивным. Имеет максимальную скорость адаптации, используя алгоритм последовательного</p>

<p>координат центра тяжести нелинейной фигуры осуществляется приближенным методом с заданной точностью.</p>	<p>обучения либо за счет модификации замены дельта-правила расчетными формулами в случае линейных функций принадлежности терм-можества входных лингвистических переменных.</p>
---	--

Контрольные вопросы

1. Биологический нейрон.
2. Искусственный нейрон.
3. Персептрон Розенблатта.
4. Искусственные нейронные сети без обратных связей.
5. Нелинейные активационные функции: Гаусса, колоколообразные, сигмоидные, степенные и т.д.
6. Процедура Уидроу-Хоффа.
7. Модифицированный метод наименьших квадратов.
8. Метод последовательного обучения.
9. Области применения искусственных нейронных сетей.
10. Алгоритм обратного распространения ошибки.
11. Генетические алгоритмы. Основные понятия.
12. Работа генетического алгоритма.
13. Адаптивный нейрон.
14. Достоинства и недостатки нечетко-нейронного управления.
15. Классификация нейронных сетей.
16. Адаптивный линейный нейрон (адалайн).

Заключение

В учебном пособии рассмотрены методы нечеткого, нейронного и нейро-нечеткого управлений недетерминированных объектов, что является общим для них. Недостатком нечеткого управления является отсутствие адаптации нечетких регуляторов к внешним воздействиям, а достоинством - прозрачность в их работе. Достоинством нейронного и нейро-нечеткого (нечеткого нейронного) управления является способность нейронной сети к обучению и возможность коррекции настроек нечеткого нейронного регулятора к изменяющимся внешним воздействиям, а недостатком – отсутствие прозрачности работы нейронной сети и исключает предшествующий опыт работы сети. Показано, что нечеткое управление обладает высоким быстродействием. В случае, когда адаптацией можно пренебречь данное управление недетерминированными объектами возможно использовать в системах реального времени. Показано, что нейронное и нейро-нечеткое (нечеткое нейронное) управление также применимо в системах реального времени, если адаптация регулятора достигается применением нейрона с последовательным методом обучения.

ЛИТЕРАТУРА

1. Федеральный закон РФ от 29 декабря 2012 г. № 273-ФЗ «Об образовании в РФ».
2. Комарцева Л.Г., Максимов А.В. Нейрокомпьютеры: Учеб. пособие для вузов. – 2-е изд., перераб., и доп. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. – 400 с.
3. Сорокин С. Системы реального времени / СТА – современные технологии автоматизации. № 2, 1997.– С. 22–29.
4. Рутковская Д, Пилиньский М., Рутковский Л, «Нейронные сети. Генетические алгоритмы и нечеткие системы» 2006. – 452 с.
5. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. – М.: Горячая линия – Телеком, 2007. – 284 с.

6. Штовба С. Д. «Проектирование нечетких систем средствами Matlab» 2007. – 288 с.
7. Гостев В.И. Проектирование нечетких регуляторов для систем автоматического управления. – СПб.: БХВ-Петербург, 2011. – 416 с.
8. Матушкин Н.Н., Хижняков Ю.Н. Нечеткий регулятор напряжения бесщеточного синхронного генератора с демпферной обмоткой // Вестник Казанского государственного технического университета им. Туполева А.Н. 2011. – № 1. – С. 61–67.
9. Леготкина Т.С., Хижняков Ю.Н. Модификация метода центроида // Вестник Ижевского государственного технического университета. – 2011. – №1. – С.122–125.
10. А.Ф. Бермант, И.Г. Абрамович. Краткий курс математического анализа. – 8-испр. и доп. – М.: Наука, 1973. – 720 с.
11. Смирнов В.А., Хасанова А.А. Особенности реализации системы управления на нечеткой логике. Изв. Челябинского научного центра, вып. 4 (21), 2003.– С. 33–38.
12. Терехов В.А. Нейросетевые системы управления. – М.: Высшая школа, 2002. – 183 с.
13. Каллан Р. Основные концепции нейронных сетей. : Пер. с англ. – Издательский дом «Вильямс», М.: 2001. – 287 с.
14. BreinMaker, California Scientific Software, Nevada City, CA95959, USA.
15. Evolver – the Genetic Algorithm Problem Solver, Axcelis, inc, 4668 Eastem Avenue N, Seattle, WA 98103, USA.
16. Хижняков Ю.Н., Южаков А.А. Нейро-нечеткий регулятор частоты газотурбинного двигателя. Приборы. – 2010. – №5. – С.17–21.
17. Леготкина Т.С., Данилова С.А. Методы идентификации систем: учеб. пособие / Т.С. Леготкина. – Пермь: Изд-во Перм. гос. техн. ун-та, 2008. – 123 с.
18. Осовский С. нейронные сети для обработки информации / Пер. с польского И.Д.Рудинского. – М. Финансы и статистика, 2004. – 344 с.