

Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа проектной деятельности и инноваций в промышленности

Доброходов Я.И., Тисенко В.Н.

Нечеткая логика Л.Заде

(материалы для самостоятельной работы по дисциплине «Логика»)

Санкт-Петербург

2026

ОГЛАВЛЕНИЕ

ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ НЕЧЁТКИХ МНОЖЕСТВ ЗАДЕ	3
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20

ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ НЕЧЁТКИХ МНОЖЕСТВ ЗАДЕ

Основание нечёткой логики Заде (НЛ) было положено профессором Лотфи А. Заде в своей статье "Нечёткие множества". Публикацией этой статьи в 1965 году он открыл новую область в теоретических исследованиях, которые привели в последние годы к настоящему взрыву в направлении практического применения НЛ, особенно в Японии /1,2/.

Отметим, что в дальнейшем непрерывнозначные логики, т.е. логики, в которых истинностный функционал принимает значения сплошь из некоторого промежутка, будем называть *непрерывными, бесконечнозначными, нечёткими, размытыми*.

Вообще говоря, впервые в мире непрерывная логика была предложена Мак-Нотоном в 1951 г. Логический формализм, построенный Мак-Нотоном, явился основой теории нечётких множеств Л.А. Заде. Можно сказать, что теория Заде - одна из возможных интерпретаций формальной системы Мак-Нотона. Однако интерпретация оказалась настолько удачной, что в настоящее время всё, связанное с нечёткими множествами, в первую очередь связывается с именем Заде, а не с именем Мак-Нотона. Мы тоже будем следовать сложившейся традиции, оставляя в стороне нравственную сторону словоупотребления "логика Заде", понимая под этим словосочетанием логику, которую использовал Заде в своей теории. Истинностный функционал в логике Заде принимает значения из отрезка $[0,1]$. Двумя годами ранее Заде в 1963 г. В.Л. Рвачев предложил свою теорию R -функций. По существу это непрерывная логика, в которой истинностный функционал принимает значения, принадлежащие всей числовой оси от $-\infty$ до $+\infty$.

Формальные системы, служащие теоретической основой обеих теорий, отличаются друг от друга. Чтобы согласиться с этим достаточно заметить, что функционалы в них принимают значения в разных множествах: в теории Заде из отрезка $[0,1]$, а у Рвачева - из бесконечного интервала $(-\infty, +\infty)$. Однако

формальные различия не делают эти логики существенно различными по их свойствам. В этом отношении они близки друг к другу. Обе теории нашли громадное число приложений. Особенно это относится к теории Заде, прикладные возможности которой поистине необъятны. “Область применений непрерывной логики расширяется не по дням, а по часам”, – говорит Дж. Бесдик, президент Североамериканского общества специалистов по обработке бесконечнозначной информации. Он объясняет причины лавинного развития теории нечётких множеств тем, что предложенный Заде подход нашел применения в таких областях, как искусственный интеллект и распознавание образов, “подчиняющихся” законам непрерывной логики.

Понимая ограниченность классической логики, Заде поставил задачу найти такой математический аппарат, который отвечал бы следующему принципу: *если люди могут принимать решения и действовать с высокой точностью в условиях, когда они не владеют точной информацией, и она не может быть выражена численно, тогда, должно быть, возможно управлять процессом, используя тот же подход, то есть в условиях нечёткой информации получать точный результат.*

Следует отметить, что в словосочетании *нечёткая логика* содержится внутреннее противоречие, связанное с многолетней привычкой к исключительной чёткости и точности классической булевой логики. В связи с этим до сих пор словосочетание *нечёткая логика* кажется слишком замысловатым и не всегда легко воспринимается. Однако, аппарат нечётких логик столь же чётко, как и аппарат любой дискретной логики, например, двузначной. Сейчас уже общепризнанно, что используя нечёткую логику, можно на основе вполне конкретных исходных данных получать вполне конкретные реальные результаты. Сам автор нечёткой логики объяснял, что термин *нечёткая* был применён к классам, не предусматривавших чётко очерченных, резких границ. Автор отдавал себе отчёт в том, внутреннее противоречие выбранного словосочетания может отталкивать желающих изучить теорию, но осознанно пошёл на это, рассчитывая, что важнее суть

явления, чем формальный термин, определяющий его. Кроме того, Л. Заде предполагал, что противоречивость терминологии может сослужить и хорошую службу, так как многие будут относиться к теории антагонистически, причём не по сути, а только из-за терминологии. Это заставит увидеть суть теории более выпукло и общо, чем если бы было в случае использования более нейтрального названия.

Описанная ситуация с названием теории, действительно, помогает понять, что теория не предусматривает каких-либо жёстко очерченных границ, она по определению полна нечётких объектов. Согласно представлениям Заде практически все человеческие высказывания полны неопределённостей и нечёткостей (например, “высокий мужчина”, “надёжный станок”, “хорошая программа испытаний” и т.д.). Все эти высказывания не совсем конкретны, и даже, если они кажутся конкретными, они всегда нечётки. Интересно отметить, что сам Заде предполагал, что нечёткая логика найдёт применение для решения прикладных задач в области биологии, психологии и лингвистики. Однако со временем он понял, что она может быть использована и для целей управления различными объектами, но даже он не думал, что сфера применения нечёткой логики будет такой обширной.

Сначала идея Заде не была воспринята серьезно и только в последние годы ситуация начала меняться, главным образом вследствие большого числа практических применений, успешно реализованных в развитых странах (США, Японии).

В нечёткой логике с термином “*логическая переменная*” можно связать любую физическую величину, для которой исследователь хочет и может иметь большее число значений, чем только *истина* и *ложь* (иными словами – “0” и “1”). При этом исследователь определяет необходимое число термов (значений любой величины, представленных не числами, а словами естественного языка), каждому из которых можно поставить в соответствие некоторое конкретное значение описываемой физической величины. Для этого значения степень принадлежности физической величины к терму будет равна “1”, а для всех

остальных значений - в зависимости от выбранной функции принадлежности. Например, при испытаниях металлорежущего станка можно ввести переменную *температура шпиндельного узла* и определить для неё такие термы как: *значительная, малая, средняя*. Определив с экспертами (специалистами по станкам) значения конкретных температур для каждого терма, можно избавиться от жёстких ограничений логики Аристотеля.

Выбранные лингвистические переменные используются в качестве *посылок и заключений* в продукционных правилах, являющихся одним из основных методов представления знаний в системах поддержки принятия решений.

В классической логике любое продукционное правило состоит из упомянутых посылок и заключений, которые объединяются посредством логических связок *И, ИЛИ*, и записывается в виде:

“если (посылка) (связка) (посылка).....(посылка), то (заключение)”.

Очевидно, что продукционное правило достаточно простое, но при этом требует наличия *полной* информации об описываемом объекте.

Нечёткие системы также основаны на правилах продукционного типа, но они существенно отличаются тем, что *в качестве посылок и заключений используются лингвистические переменные*. В результате становится возможным избежать ограничений, которые присущи классическим продукционным правилам.

В системах, использующих нечёткую логику, все входные и выходные переменные ассоциируются с нечёткими множествами. Эти множества связываются друг с другом через правила вида

“если x и y , тогда z ” (“если x или y , тогда z ”).

Правила устанавливают условные связи между входными и выходными переменными объекта, образуют базу, основывающуюся на собранных знаниях об объекте, и определяют необходимую стратегию управления.

Таким образом, нечёткая логика использует описание объекта в терминах естественного языка. Нечёткие множества конструируются путем

комбинирования значений переменной, субъективно выраженной через лингвистические термины, и функцией принадлежности (ФП) (*membership function – MBF*). ФП – это функции, которые каждому значению переменной (в границах каждого множества) ставят в соответствие значение ФП (*membership degree MBD*) от 0 до 1.

Результат такого нечёткого логического вывода является, конечно, также нечётким, который не может быть использован, например, для функций управления объектом. Поэтому необходимо провести специальные математические преобразования, позволяющие перейти от нечётких значений величин к вполне определенным, которые можно было бы использовать в системах управления и других случаях.

Описанный выше способ рассуждений позволяет исследователю легко оперировать неточными, размытыми количественными понятиями - такими, например, как "большинство", "много", "мало", "почти ноль" и т.д., которые обычно использует человек, размышляя о ситуациях реального мира.

Нечёткая логика связывает воедино принципы классической стандартной логики и теории вероятностей, так что появляется возможность учитывать различные степени неопределенности, например, в уравнениях, порождающих правила экспертной системы. Если говорить о практической работе, то нечеткая логика открывает доступ к достижениям искусственного интеллекта, хотя в настоящее время существует мнение о сложности и разочаровывающих результатах обычных подходов, которые используются сейчас в области искусственного интеллекта.

В тех случаях, когда важен опыт человек (как, например, при разработке систем искусственного интеллекта), по утверждению Л. Заде обычно легче выражать те или иные понятия неточными терминами, подобно тому, как люди объясняют те или иные вещи словесно. Сторонники нечеткой логики доказывают, что не слишком сложные программы становятся значительно меньше по объему, чем если бы они строились с использованием

традиционных методов, в некоторых случаях выигрыш может достигать нескольких порядков.

С момента, когда нечеткая логика впервые была внедрена в 1980 г. в Швеции для управления печами обжига цемента - производства, которое с точки зрения возможностей и уровня автоматизации находилось на одном из последних мест, аппаратные средства нечеткой логики начали становиться в один ряд с изделиями, пользующимися наибольшим спросом /3/. Ещё в 1992 году Масаки Тогай считал, что в ближайшем будущем в 10% различных экспертных систем будут содержаться средства нечеткой логики. Наиболее перспективными для их внедрения являются системы для бизнеса и оборудование бытовой электроники.

Главное преимущество нечеткой логики заключается в ее простоте и в том обстоятельстве, что при использовании классической логики "точность обходится дорого". По утверждению Л. Заде одной из причин, по которой японцы возглавили работы в области нечеткой логики, заключается в том, что они первыми поняли, что *во многих случаях высокая точность просто не нужна и использование нечеткой логики вполне оправдано.*

Нечёткая логика отвергает категоричные "да" и "нет" и принимает нечеткие понятия, такие как, например, "лучше", "хуже". Очевидно, что некоторое событие, подразумевающее выполнение условий A и B , произойдет с меньшей вероятностью, чем то, которое состоится при выполнении условия A или B . Иначе говоря второе сочетание "лучше" первого. На языке алгебры логики первое сочетание называется конъюнкцией, второе - дизъюнкцией, и вероятность выполнения конъюнкции двух условий меньше вероятности выполнения их дизъюнкции. Запишем это таким образом:

$$V(A \wedge B) < V(A \vee B), \quad (1)$$

где под V подразумевается любая оценка выполнения события, удовлетворяющая условиям A , B и последовательности их выполнения.

Рассмотрим другой случай и оценим выполнение некоторого события, если:

- * оно выполняется только одним условием A ;
- * оно может выполняться при выполнении либо условием A или B ;
- * возможно его выполнение при выполненном одном из условий A , B , C .

Конечно, в последнем случае мы будем иметь большую по сравнению с другими вероятность, надежность, точность и т.д. Этот пример похож на осуществление резервирования:

$$V(A) < V(A \vee B) < V(A \vee B \vee C) \quad (2)$$

Иначе говоря, дизъюнкция нескольких элементов будет иметь меньшую оценку, чем дизъюнкция тех же элементов с добавлением одного или нескольких новых элементов. Противоположным образом дело обстоит с конъюнкцией:

$$V(A) > V(A \wedge B) > V(A \wedge B \wedge C) \quad (3)$$

Нужно отметить и другие правила логики Заде (или нечеткой):

- * при увеличении оценки выполнения условия увеличивается оценка всей системы

$$V(B) < V(C) \implies V(A \vee B) < V(A \vee C);$$

$$V(A \wedge B) < V(A \wedge C) \quad (4)$$

- * при увеличении суммы оценок (элементов) выполнения условий увеличивается общая оценка

$$V(A) + V(B) > V(D) + V(C) \implies$$

$$V(A \vee B) > V(D \vee C); \quad V(A \wedge B) > V(D \wedge C) \quad (5)$$

- * оценка конъюнкции по методике Заде

$$V(A \wedge B) = \min [V(A); V(B)]; \quad (6)$$

- * оценка дизъюнкции:

$$V(A \vee B) = \max [V(A); V(B)]. \quad (7)$$

Итак, познакомившись в общих чертах, с аксиоматикой логики Заде, попробуем определить её конкретные отличия от булевой двузначной логики.

Основным отличием является употребление в качестве оценок событий не “0” и “1”, а *любых чисел между ними*. Оценки чаще всего носят сравнительный характер.

Формальный аппарат логики Заде, безусловно, делает шаг вперед, оставаясь в рамках двузначной логики, но он не соответствует нашей интуиции во многих прикладных задачах, то есть сужает круг своего действия.

Кроме того, в данном методе не различаются связи между элементами, он не различает число элементов системы, так как ориентирован либо на максимум, либо на минимум.

С учетом вышесказанного теорией Заде предписывается *реализация трех этапов*:

- *фазификация или переход к нечёткости (fuzzification)*, когда точные значения входных переменных преобразуются в значения лингвистических переменных посредством применения определенных функций принадлежности;
- *оценка правил (rule evaluation)*, когда определяются продукционные правила, связывающие лингвистические переменные. Совокупность этих правил описывает связи между входными переменными и выходными;
- *дефазификация (defuzzification)*, когда осуществляется переход от нечётких значений величин к вполне определенным физическим параметрам.

В силу того, что теория нечётких множеств мало известна в широких кругах специалистов, занимающихся практическими прикладными задачами моделирования и управления, подробнее рассмотрим каждый из отмеченных этапов, схема реализации которых представлена на рис. 1.

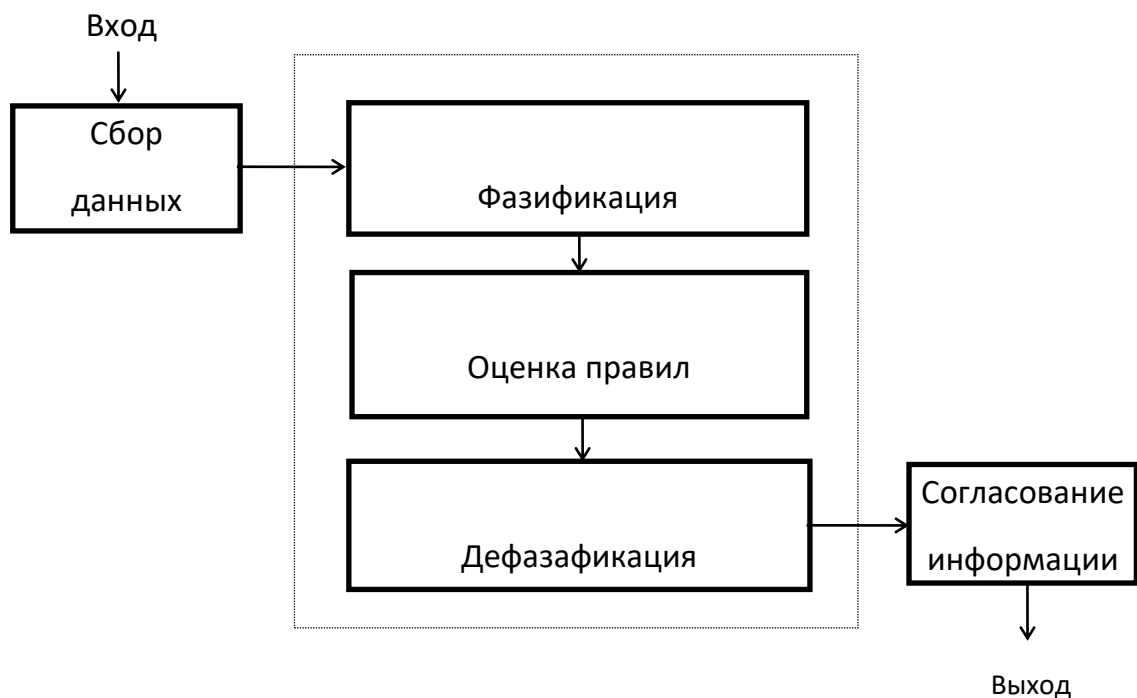


Рис.1

Схема системы управления с использованием нечеткой логики Заде

На этапе *фазификации* каждому из термов всех лингвистических переменных необходимо поставить в соответствие точные физические значения. Согласно теории нечётких множеств каждому значению, например, температуры из указанного диапазона может быть поставлено в соответствие некоторое число от 0 до 1, которое определяет *степень принадлежности* данного физического значения температуры к тому или иному терму лингвистической переменной “температура” (рис.1). Конкретное определение степени принадлежности производится экспертным методом.

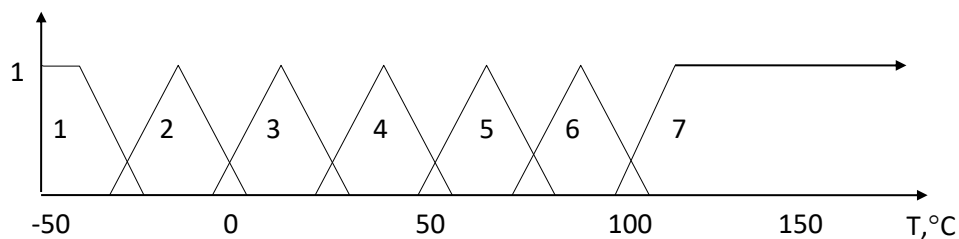


Рис.2

Ранжирование температуры и функция принадлежности (1- чрезвычайно холодно, 2- очень холодно, 3- холодно, 4- прохладно, 5- тепло, 6- горячо, 7- очень горячо)

На рис.2 ось X представляет количественную оценку входных и выходных значений (в нашем случае – температуры) (*область определения*), а ось Y – область значений функции принадлежности (*membership degree – MBD*). Пересечения областей нечётких множеств называются перекрытиями (*overlap*).

Количество термов в переменной принято выбирать из ряда 3 – 7. Это обусловлено тем, что при количестве термов, равном 3, можно определить соответствующую лингвистическую переменную через максимальное, минимальное и среднее значения. В то же время количество термов, равное 7, объясняется возможностями эксперта хранить и обрабатывать информацию. Хотя, конечно, увеличение количества термов приводит к повышению точности расчетов, но и существенно усложняет процедуру моделирования.

На этом же этапе производится выбор функций принадлежности.

На этапе *дефазификации* осуществляется процесс обратного преобразования численных значений из среды нечётких множеств в численные значения “реального мира”.

Таким образом, *первый шаг* в построении контроллера нечёткой логики - сопоставление лингвистических терминов каждому из контролируемых физических параметров. Цель - разделить значения каждого параметра на области и поименовать их. В таблице 1 приводится пример такого деления

на области для температуры в соответствии с рис.2. Обычно это разделение зависит от мнения разработчика и/или эксперта, либо отражает знания, полученные о системе.

С каждой из областей сопоставляется функция, через которую будет определяться значение ФП. Сопоставление ФП с областями приводит к образованию нечётких множеств.

Ту же процедуру надо выполнить, чтобы определить нечёткие множества для остальных переменных (например, давления и величины потока воды). Остальные таблицы здесь не представлены, но результаты определения нечетких множеств представлены на рис.3 и рис.4.

Таблица 1

Разделение значений температуры на области

Нечёткие множества	Лингвистические термины	Области температуры (°C)
1	чрезвычайно холодно	-50 до -10
2	очень холодно	-15 до 0
3	холодно	-5 до 20
4	прохладно	15 до 50
5	тепло	45 до 75
6	горячо	70 до 110
7	очень горячо	105 до 175

В зависимости от рассматриваемого процесса и используемых параметров могут быть применены различные ФП. Большинство применяемых на практике функций линейны или являются комбинациями линейных функций, так как вычисления при этом простые и не требуют много памяти.

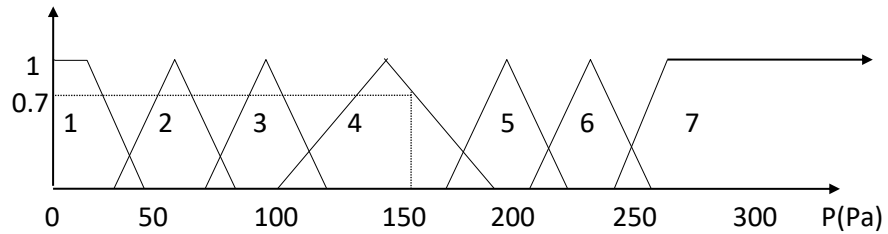


Рис.3

Ранжирование давления и функции принадлежности

(1 – очень низкое, 2 – низкое, 3 – среднее, 4 – высокое, 5 – очень высокое)

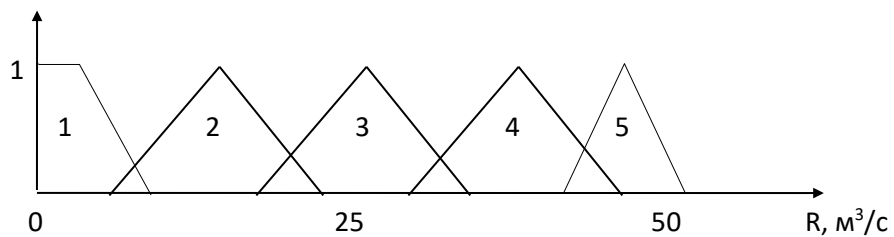


Рис 4

Ранжирование величины потока и функции принадлежности

(1.- очень слабое; 2 - слабое; 3 - среднее; 4 - сильное; 5 – очень сильное)

Следующий шаг в проектировании системы управления – разработка схемы управления через определение основных правил работы. Эти правила должны отражать знания, полученные об управляемом объекте, и определять необходимую стратегию управления. Проиллюстрируем это на нашем примере, предположив для простоты, что система функционирует правильно, базирясь на 4-х основных правилах:

1. **если** темп. = чрезв.-холодно **и** давление=среднее, **тогда** поток воды =

слабый

2. **если** темп. = холодно **и** давление = среднее, **тогда** поток воды = средний
3. **если** темп. = прохладно **и** давление = среднее, **тогда** поток воды = сильный
4. **если** темп. = теплый **и** давление = среднее, **тогда** поток воды = очень сильный

На этой стадии процесс проектирования контроллера нечёткой логики полностью завершен. Другими словами, нечёткие множества определены и правила сформулированы. Эти шаги могут быть выполнены за достаточно короткий промежуток времени с использованием специального пакета программ. Далее такой специальный пакет программ может генерировать программу, которая будет выполнять алгоритм управления с НЛ. Затем программа загружается в контроллер и запускается на исполнение. В процессе работы программа выполняет все три описанные выше фазы процесса НЛ: фазификацию, оценку правил, дефазификацию. Краткое описание всех этих операций для рассматриваемого примера приводится ниже.

Фазификация

Процесс фазификации выполняется в процессе работы программы и состоит в присваивании значений функции принадлежности (от 0 до 1) значениям параметров реального мира. Из рис. 2 видно, что значение температуры 50°C принадлежит множеству “прохладно” со степенью 0,2 (20%) и множеству “тепло” со степенью 0,5 (50%). Из рис.3 видно, что значение давления 150 Pa принадлежит множеству “среднее” со степенью 0,7 (70%).

Оценка правил

В большинстве случаев в НЛ применяются правила следующего вида:

если [условие A и условие B] тогда [результат C]

Здесь оператор конъюнкции (**и**) работает как min-функция, оставляя меньшее из значений ФП, полученных после фазификации. В нашем примере

значению температуры 50°C соответствуют множества “прохладно” и “тепло”, а значению давления 150 Па, соответствует множество “среднее”. При таких значениях реальных параметров используются правила 3 и 4, а правила 1 и 2 не принимаются в рассмотрение.

Оценка правила 3: **Если** температура “прохладная” и давление “среднее”, **тогда** поток воды “сильный”.

Так как в рассматриваемом случае температура = “прохладная” со степенью 0,2, а давление = “среднее” со степенью 0,7, то правило оценивается числом 0,2, полученным как $\min(0,2; 0,7)$.

Тогда в соответствии с этой оценкой выходная переменная (водный поток) принимает значение “сильный” со степенью 0,2.

Оценка правила 4: **Если** температура “теплая” и давление “среднее”, **тогда** поток воды “очень сильный”.

Так как в рассматриваемом случае температура = “теплая” со степенью 0,5, а давление = “среднее” со степенью 0,7, то правило оценивается числом 0,5, полученным как $\min(0,5; 0,7)$.

В соответствии с этой оценкой, выходная переменная (водный поток) принимает значение “очень сильный” со степенью 0.5.

Таким образом, на этой стадии правила 3 и 4 оценены, но абсолютного решения нет до тех пор, пока не присвоено значение выходной переменной. Основные выводы делаются в результате процесса дефазификации.

Дефазификация

Процесс дефазификации заключается в расчете единственного “реального” значения для выходной переменной на основании “нечетких” значений, полученных в результате оценки правил. Существуют три основных способа проведения дефазификации:

* метод максимальной дефазификации или метод наибольшего значения (принимается во внимание только максимальное число, полученное в результате оценки правил).

* метод центра масс или метод центра максимума (принимаются во

внимание все значения и расчет производится по формуле центра масс):

$$\frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i},$$

где f_i – i -ое значение ФП для выходной переменной;

x_i – значение выходной переменной, соответствующее максимальному значению ФП для данного множества;

n – порядковый номер соответствующего множества.

* метод центра площади или метод центроида (метод сходен с предыдущим за исключением того, что в нём используется интегрирование вместо суммирования). Иными словами, в этом методе значение выходной величины является координатой центра тяжести фигуры, получаемой усечением характеристических функций выходной величины полученными значениями их степеней принадлежности.

Рассмотренный практический пример применения нечетких логик показывает, что для решения задачи достаточно выполнить основные шаги по определению входных и выходных переменных, соответствующих нечетких множеств и основных правил. Нет необходимости в использовании математических уравнений и классического анализа. Единственное, что требуется - достаточно хорошее знание системы для определения адекватных нечетких множеств для каждого параметра и заполнения таблиц правил для осуществления стратегии управления.

Анализ состояния дел с применением новых математических средств моделирования сложных технических систем показывает, что в то время, как управляющие системы становятся все более и более сложными, спрос на применение методов НЛ постоянно растет. Это в особенности верно в тех случаях, когда *система, подлежащая управлению, не имеет точной математической модели или является нелинейной*. Для таких систем применение подобных методов из области искусственного интеллекта дает разумные, эффективные решения.

НЛ была определена как один из первых полезных продуктов искусственного интеллекта в области управления. Японцы применяют НЛ даже для тех систем, для которых годятся и обычные методы. Это происходит в следствие того, что цикл проектирования системы управления с НЛ требует меньше времени, так как аналитический этап заменен этапом применения полученных знаний об объекте управления. Другая причина в том, что линейная ФП, которая обычно применяется в управлении, требует очень мало памяти при работе с контроллером НЛ, а контроллер производит минимальные вычисления. В результате простые 8-ми и 16-ти разрядные микроконтроллеры хорошо подходят для большого числа применений.

Основная трудность НЛ состоит в нехватке (ограниченности) методологии формального проектирования. Другая трудность – результирующая система не является аналитической, не остается возможности для какого-либо анализа. Оптимизация не может быть проведена с использованием производных и определением минимумов и максимумов, так как нет математических уравнений. Однако эти трудности не мешают НЛ доказывать свою жизнеспособность на большом количестве практических применений, преимущественно в Японии.

В заключении отметим, что в настоящее время нечёткое управление на основе логики Заде реализуется специальными нечёткими контроллерами, в основе которых лежит так называемая “машина нечётких выводов” /93/, которая реализует нечёткий вывод типа “Если А, то В”, соответствующий этапу принятия решения. Базовая архитектура нечёткого контроллера, выполняющего r нечётких выводов, соответственно содержит r машин нечётких выводов. Кроме этого, контроллер содержит нечёткую память, в которой хранится база знаний и схема объединения результатов выводов. К концу 80-х годов в Японии был уже налажен выпуск специализированных нечётких контроллеров, выполненных по технологии СБИС.

Однако нечёткое управление может выполняться и с помощью обычных или специализированных логико-арифметических контроллеров, в которых

нечёткие выводы выполняются путём вычисления характеристических значений выходной логической переменной через характеристические значения входных лингвистических переменных по логическим формулам, использующим логические операции “И”, “ИЛИ”, “НЕ”. В частности, американская фирма Intel, понимая своё отставание от освоивших выпуск нечётких контроллеров японцев, пошла по этому пути. Фирма предоставила разработчикам средство разработки приложений на базе хорошо освоенных собственных контроллеров, но с использованием технологии нечёткости. Создание такой технологии позволило избежать значительных затрат на создание собственных нечётких контроллеров, а разработанная в середине 80-х годов под руководством профессора Ганса Циммермана система FuzzyTECH получила широкое распространение в США, Европе и даже в Японии. Для работы системы FuzzyTECH разработан специальный аппаратно независимый объектно-ориентированный язык FTL (Fuzzy Technology Language). Он имеет открытую архитектуру и может быть расширен в зависимости от потребностей пользователя. В настоящее время существует версия системы FuzzyTECH для MS-Windows 3.1.

ЗАКЛЮЧЕНИЕ

В настоящее время намечается существенное возрастание интереса многих разработчиков информационных технологий и искусственного интеллекта к современным нечетким логикам применительно к задачам, относящимся к классу задач, рассматриваемых в условиях неопределенностей. Нечеткая логика Заде показала себя очень удобным средством формализации законов управления объектами, относительно которых отсутствует их четкое и достаточно полное понимание.

Достаточно многолетнее практическое применение нечеткой логики Заде показало простоту и удобство ее использования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Инамори К. Страсть к успеху. Японское чудо / Пер. с англ. / Ростов-на-Дону, Феникс, 1998. - 301 с.
2. Половко А.М. Принципы построения абсолютно надёжных технических устройств. - Л.: ЛДНТП, 1990. - 20 с.
3. Тисенко В.Н. Особенности организации экспериментальных работ на первых этапах внедрения ГПС. // Опыт применения малых и микро-ЭВМ в АТК и ГПС металлообработки: Тезисы докладов междунар. конфер./ НИИТ КФ.- Казань, 1988.- С. 31-35.
4. Половинкин А.И. Основы инженерного творчества. - М.: Наука, 1964. - 448 с.