

**УПРАВЛЕНИЕ
ДЕТЕРМИНИРОВАННЫМИ СИСТЕМАМИ**

УДК 658.012.011.56

**СИНТЕЗ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ С ИСПОЛЬЗОВАНИЕМ
МЕХАНИЗМОВ ОБРАБОТКИ НЕЧЕТКОЙ ИНФОРМАЦИИ***

© 2000 г. В. Н. Захаров, П. С. Кантор, Д. Г. Ткаченко

Москва, ВЦ РАН, МОКБ “МАРС”

Поступила в редакцию 20.11.99 г.

Рассматриваются средства преодоления неопределенности входной информации при составлении формализованных описаний работы логических контроллеров, используемых на исполнительном уровне интеллектуальных систем логического управления. Предложена методика построения описаний автоматных моделей с применением средств нечеткой логики при наличии неопределенной информации и данных, полученных от экспертов. Приведены примеры работы алгоритмов обработки нечеткой информации и построения формальных моделей логических контроллеров.

Введение. Логические контроллеры, согласно [1], не принадлежат к классу интеллектуальных управляющих систем, поскольку при формировании управляющих воздействий в их структурах не используются ни базы знаний, ни механизмы вывода на знаниях, традиционные для любых интеллектуальных систем управления. Однако логические контроллеры могут применяться на исполнительном уровне в структурах интеллектуальных систем логического управления, под которыми подразумеваются управляющие системы со специальным образом организованной многоуровневой архитектурой, на нижних уровнях иерархии которых используются формальные логические модели. Управляющие структуры систем такого класса строятся в соответствии с принципом “более высокому уровню в иерархической структуре соответствует более высокая степень интеллектуальности и наоборот” [1]. Поскольку понятие “интеллектуальность” в общепринятом смысле является не формализуемым, обычно под “интеллектуальностью системы” понимается способность системы работать с базой внешних событий (ситуаций) в условиях не полной определенности поступающей входной информации. А под “степенью интеллектуальности” – степень развитости используемых средств преодоления неопределенности на основе знаний закономерностей проблемой среды или внешнего мира, в котором функционирует данная управляющая система. Таким образом, исполнительный уровень (уровень формальных моделей) относят к уровням с нулевой степенью интеллектуальности, т.е. к уровням, неспособным к работе с неточной и с неопределенной информацией. На исполнительном уровне в интеллектуальных системах логическо-

го управления используются, как правило, логические контроллеры или управляющие устройства, организованные с применением логических моделей (комбинационных схем или схем с памятью). Обработка неопределенной информации осуществляется на самых верхних уровнях иерархической структуры с использованием таких средств, как приближенный вывод, правдоподобные рассуждения, мягкие вычисления, привлечение средств обучения и т.п.

При логическом синтезе исполнительных устройств приходится также сталкиваться с недоопределенностью используемой информации. В [2] в качестве средств преодоления неопределенности исходной информации был предложен специальный язык продукционного типа [3], названный языком секвенций. Он удобен при построении формальных моделей логических устройств, однако требует наличия законченных вербальных описаний их работы. Между тем неопределенность исходной информации и даже отсутствие требуемых данных могут проявиться как раз на этапе построения вербальных описаний работы проектируемых устройств.

В данной статье анализируется именно этот случай и предлагается подход к решению проблем синтеза логических контроллеров, основанный на использовании теории нечетких множеств и методов обработки нечеткой информации.

1. Постановка задачи. Методология проектирования логических устройств на базе автоматных моделей известна с давних времен и в общих чертах представляет собой следующую совокупность процедур (см., например, рис. 1 в [2]). На основании исходных данных строится вербальное описание работы проектируемого устройства, представленное в виде текста на естественном языке. В дальнейшем это описание формализует-

*Работа выполнена при финансовой поддержке РФФИ (грант 99-01-00150а).

ся. Результатом этого процесса служит построение формальной модели проектируемого устройства. В случае использования модели конечного автомата в качестве формальной модели рассматривается автоматная таблица или автоматный граф. Далее выполняется анализ полученной формальной модели, ее минимизация, кодирование и переход к таблицам, задающим функции выходов и внутренних переменных (в случае автомата с памятью). Следующий этап состоит в построении компактных аналитических выражений для полученных функций (систем функций), по которым и производится построение логической структуры, реализующей исходное описание проектируемого устройства. Заключительный этап синтеза – построение реальной структуры проектируемого устройства управления, которая отличается от полученной на предыдущем этапе логической структуры учетом реальных ограничений, присутствующих реальным элементам структуры.

Переход от вербального описания к формальной модели связан с кодированием числовой информации, содержащейся в вербальном описании. При двоичном кодировании число кодовых переменных n выбирается из условия

$$n = \begin{cases} k & \text{при } N = 2^k \\ \lceil \log_2 N \rceil + 1 & \text{при } N \neq 2^k, \end{cases} \quad (1.1)$$

где N – некоторое число, которое требуется закодировать (например, число состояний автомата), а $\lceil A \rceil$ – целая часть числа A .

В рамках формальной модели должны существовать средства для отображения недоопределенной информации (логических функций). В случае автоматов таким средством служат автоматные таблицы, в которых имеются прочерки. Для перехода к аналитическому представлению недоопределенных булевых функций в [2] предложено использовать упомянутый выше язык секвенций, представляющий собой частный случай продукционных правил, в левых и правых частях которых присутствуют булевы функции.

Однако, как это отмечалось во введении, при синтезе логических контроллеров недоопределенность используемой информации может возникнуть и раньше, а именно на этапе составления вербального описания, для которого может не доставать требуемых для кодирования при построении автоматных таблиц числовых данных. В ряде случаев построить вербальное описание можно лишь с привлечением качественной информации, непосредственный перевод которой в количественные данные затруднен или невозможен. Для преодоления указанной неопределенности в данной статье предлагается использовать методы представления и обработки нечеткой информации, ис-

пользуемые при разработке нечетких контроллеров.

2. Концептуальная схема обработки нечеткой информации. Одним из наиболее распространенных способов представления нечеткой информации являются нечеткая логика и лингвистические переменные [4–6]. Под лингвистической переменной (ЛП) понимается переменная, значениями которой являются не числа, а слова или словосочетания из некоего естественного или искусственного языка. Множество таких значений ЛП образует терм-множество. Множество значений аргументов ЛП называется обсуждаемым универсумом. ЛП можно представить графически, если каждому значению обсуждаемого универсума поставить в соответствие значение от 0 до 1 некой характеристической функции, называемой функцией принадлежности и выражающей степень уверенности эксперта, относящего значение данного аргумента определенному элементу терм-множества. Организация процесса обработки нечеткой информации начинается с процесса построения базы нечетких правил, с этой целью и выделяются лингвистические переменные, существенным образом влияющие на процесс управления. В [4] приведена концептуальная схема обработки нечеткой информации (КСОНИ), на базе которой строится структура нечеткого контроллера. Там же перечислены основные этапы получения точных выходных значений на базе обработки с помощью КСОНИ значений входных переменных: этап фазификации, этап вычислений альфа-значений, этап нечеткого вывода, этап дефазификации. На рис. 1 приведена структура КСОНИ из [4] в составе нечеткого контроллера, реализующая указанные четыре этапа.

Прежде чем приступить к выполнению первого этапа (фазификация), под которым подразумевается процедура перевода четкой (числовой) информации с датчиков, контролирующих состояние объекта или процесса управления, в нечеткие лингвистические значения, необходимо выполнить следующие не указанные в [4] шаги.

Шаг 1. Определить все необходимые входные и выходные переменные, оказывающие существенное влияние на процесс нечеткого управления.

Шаг 2. Каждой переменной поставить в соответствие лингвистическую переменную.

Шаг 3. Определить возможные пределы изменения всех измеряемых величин, включая выходные переменные (определить для каждой лингвистической переменной обсуждаемый универсум).

Шаг 4. Для каждой лингвистической переменной задать терм-множество лингвистических значений.

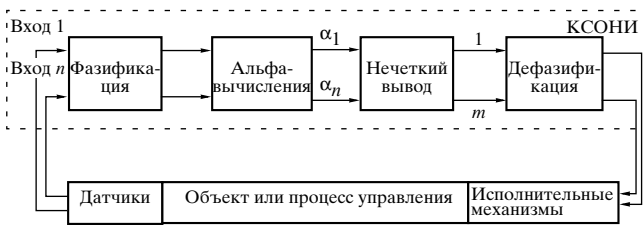


Рис. 1. Блок КСОНИ в структуре нечеткого контроллера.

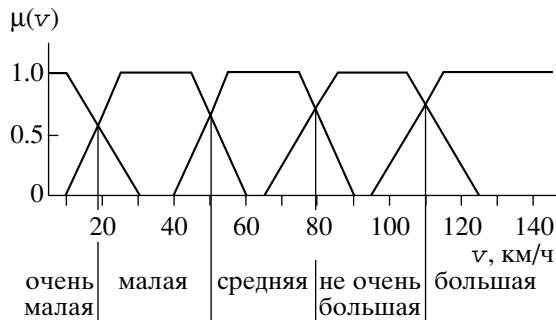


Рис. 2. Значения терм-множества для лингвистической переменной "скорость автомобиля".

Шаг 5. Определить значения функций принадлежности измеряемых величин элементам терм-множеств.

После выполнения этих пяти шагов можно приступать к реализации этапа фазификации. Наиболее сложным для реализации является этап нечеткого вывода, представляющий собой расширенный модус поненс, схема которого приведена в [4]:

Посылка 1: Если X есть A , то Y есть B

Посылка 2: X есть A'

Следствие: Y есть B' , где $B' = A' \circ (A \rightarrow B)$ – композиционное правило вывода.

В приведенной схеме нечеткого вывода величина A и B – элементы терм-множества определенной лингвистической переменной, а величина A' и B' – нечеткие приближения к величинам A и B . Использование этой схемы приводит к необходимости хранения функций принадлежности антецедентов и их обработки при реализации процедур нечеткого вывода. В результате получаются функции принадлежности консеквентов. На последнем этапе определяются четкие значения выходных переменных, определяющих воздействия на исполнительные механизмы. Основными методами дефазификации являются метод центра тяжести, метод центрирования и метод середины максимума [4].

3. От нечеткого контроллера к логическому.

Перечисленные процедуры, необходимые при разработке нечеткого контроллера, могут оказаться

полезными для уточнения задания на проектирование обычного логического контроллера путем лингвистической аппроксимации не полностью определенных исходных данных на его разработку. Однако в случае логического контроллера операции нечеткого вывода и дефазификации становятся ненужными. Обсуждаемый универсум разбивается построенными функциями принадлежности терм-множеств на отрезки, каждому из которых соответствует четкое правило логической продукции (секвенции [2]) вида $\Phi \vdash \Psi$, где Φ – интервал универсума, относящийся к определенному элементу терм-множества. На рис. 2 дан пример такого разбиения для скорости движения автомобиля, для которого обсуждаемый универсум составляет отрезок от 0 до 200 км/ч. Терм-множество состоит из пяти элементов: {очень малая, малая, средняя, не очень большая, большая}.

Этот рисунок может служить основой для работы фазификатора, преобразующего конкретные величины скоростей, считываемых со спидометра, в соответствующие лингвистические значения терм-множества. Для нечеткого контроллера информация дополняется данными альфа-вычислений и соответствующая пара (элемент терм-множества, значение функции принадлежности) поступает в блок нечеткого вывода. Для логического контроллера достаточно знания элемента терм-множества, которое поступает на вход правила логической продукции и непосредственно формирует значение выходного управляющего воздействия.

Общая схема процесса разработки логического контроллера при отсутствии точных данных для формализации условий его работы выглядит следующим образом.

На основе имеющихся исходных данных составляется вербальное описание работы логического контроллера. После этого выполняются описанные выше шаги 1–5, для выполнения которых привлекаются знания экспертов, сообщающих значения функции принадлежности для различных элементов терм-множеств. Далее осуществляются следующие дополнительные шаги.

Шаг 6. Определение всех возможных наборов лингвистических значений входных и выходных переменных.

Шаг 7. Кодирование всех наборов, полученных на шаге 6.

Шаг 8. Построение автоматной таблицы (соответствие выходных наборов входным).

Шаг 9. Переход к построению структуры логического контроллера, имеющей вид, показанный на рис. 3.

Как видно из рис. 3, логический контроллер получается из структуры, показанной на рис. 1, заменой блока КСОНИ на логический блок, состоящий из аналогово-цифрового преобразова-

теля (АЦП), автомата А (с памятью или без памяти) и формирователя управляющих воздействий (ФУВ). Существо описанного подхода к построению логического контроллера состоит в том, что используемая нечеткая аппроксимация неполных данных о работе системы заменяется в дальнейшем логической схемой (превращается в четкую схему управления). Проиллюстрируем особенности реализации описанных выше шагов на примере разработки специализированного контроллера, выполняющего одну из важных функций аппарата искусственной вентиляции легких.

4. Пример разработки логического контроллера с неточными исходными данными. Пусть требуется разработать управляющее устройство процессов искусственной вентиляции легких, решающее одну из важнейших задач – задачу перевода пациента на самостоятельное дыхание. Исходные медицинские данные для разработки такого контроллера следующие.

Замещение утраченного или ослабленного самостоятельного дыхания (СД) с помощью искусственной вентиляции легких (ИВЛ) осуществляется в течение достаточно долгого времени и является одним из существенных компонентов респираторной поддержки (РП). Появление целого ряда признаков, свидетельствующих о начале восстановления этой жизненно важной функции организма, приводит к необходимости принятия врачом решения о прекращении ИВЛ.

Процесс прекращения непрерывной ИВЛ является длительным (занимающим нередко до 40% времени от общей длительности РП) и трудоемким. Он требует постоянного контроля за состоянием пациента и динамикой показателей, отражающих функциональное состояние кардиореспираторной системы.

Понятное стремление врача к более быстрому прекращению ИВЛ нередко приводит к обратному результату: вследствие развития декомпенсации сердечно-сосудистой и дыхательной систем в ранние сроки перевода на самостоятельное дыхание возникает необходимость возврата к длительной ИВЛ.

Эти данные медицинских наблюдений могут служить основой для реализации с помощью аппарата ИВЛ следующей тактики перевода пациента на самостоятельное дыхание. Пациента с нарушенной функцией дыхания, подключенного к аппарату ИВЛ, можно попытаться перевести на самостоятельное дыхание, если все датчики, контролирующие процесс вентиляции, свидетельствуют о нормальном дыхании. При этом нельзя производить резкое отключение пациента от аппарата. А нужно немного понизить давление подаваемой пациенту газовой смеси в том случае, если его дыхание нормально, и проанализировать, не ухудшаются ли показатели самочувствия

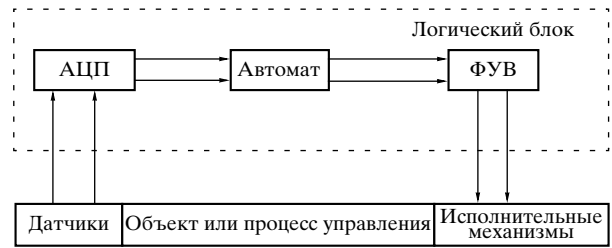


Рис. 3. Концептуальная структура логического контроллера.

пациента. Если ухудшаются, то вновь вернуться к нормальному давлению и повторить попытку через некоторое время. Если показатели самочувствия не ухудшаются, а все параметры дыхания находятся в норме, снова понизить давление и т.д. до полного отключения аппарата.

Как следует из этого вербального описания работы контроллера, исходных данных явно не достаточно для формализации условий его работы. Не ясно, какие параметры следует учитывать при определении нормального дыхательного процесса и что значит, что контролируемые величины находятся “в норме”? Необходимо уточнить все эти детали путем лингвистической аппроксимации условий работы контроллера в процессе согласования условий работы проектируемого устройства на этапе разработки, традиционно называемом “разговором с заказчиком” [7]. Под заказчиком подразумевается лицо, поручающее вести разработку системы управления другому лицу, называемому “разработчиком”. В нашем случае заказчиком является врач – специалист в области процессов искусственной вентиляции легких, который и будет одновременно исполнять функции эксперта. Процесс передачи от заказчика разработчику необходимых для разработки системы управления сведений подробно описан в [7]. Перейдем к подробному изложению описанных выше шагов.

Входные и выходные переменные, оказывающие существенное влияние на процесс управления ИВЛ, в соответствии с шагом 1 определяются по согласованию с заказчиком и содержат следующий набор. Входные неуправляемые переменные: x_1 – частота сердечных сокращений, x_2 – артериальное давление, x_3 – содержание гемоглобина, x_4 – частота собственного дыхания. Выходная переменная (она же – управляемый параметр) y – давление газовой смеси, подаваемой пациенту. Для выполнения следующих шагов вернемся к понятию лингвистическая переменная (ЛП).

На рис. 4а дан пример ЛП, называемой “частотой сердечных сокращений” (ЧСС), построенной по данным эксперта-врача и используемой в приведенном далее примере. Множество аргументов ЛП, соответствующих обсуждаемому универсу-

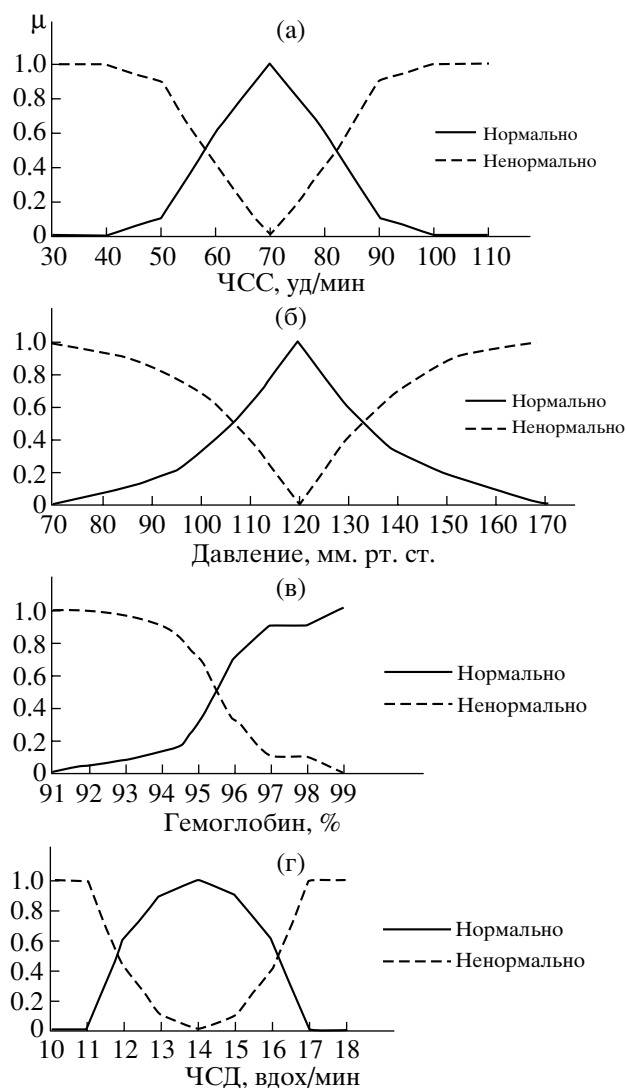


Рис. 4. Графики зависимости функции принадлежности $\mu(x)$ термам “нормально”, “ненормально” от частоты сердечных сокращений (а), артериального давления (б), содержания гемоглобина (в) и частоты собственного дыхания (г).

му, ограниченно интервалом от 30 до 110 ударов в минуту. Терм-множество ЛП ЧСС представлено двумя значениями: “нормально”, “ненормально”. Каждому значению аргумента поставлено в соответствие два значения функции принадлежности этого значения лингвистическому значению “нормально” и “ненормально”. Например частота 50 ударов в минуту принадлежит лингвистическому значению “нормально” с мерой уверенности 0.2 и значению “ненормально” – с мерой уверенности 0.8.

Построение функций принадлежности – наиболее ответственная и важная часть моделирования процесса нечеткого управления, поскольку эти функции определяют результаты обработки

нечеткой информации, получаемой в словесной форме от экспертов. Среди существующих к настоящему времени методов построения функций принадлежности можно выделить два класса, прямые и косвенные. Прямыми методами построения функций принадлежности обычно называют такие методы, в которых значения функций принадлежности непосредственно определяются или задаются одним либо несколькими экспертами. ЛП, указанная на рис. 4а, получена соединением прямыми отрезками значений функций принадлежности, полученных от эксперта-врача, для лингвистического значения “нормально” и имеющих следующий вид:

$$\begin{aligned} \mu(30) &= 0.08, \quad \mu(40) = 0.1, \quad \mu(50) = 0.2, \\ \mu(60) &= 0.5, \quad \mu(70) = 1, \quad \mu(80) = 0.5, \\ \mu(90) &= 0.2, \quad \mu(100) = 0.1, \quad \mu(110) = 0.08. \end{aligned}$$

Значения функции принадлежности для лингвистического значения “ненормально” могут быть вычислены по формуле

$$\bar{\mu}(x_i) = 1 - \mu(x_i),$$

где $\mu(x_i)$ – значения функции принадлежности для лингвистического значения “нормально”, а $\bar{\mu}(x_i)$ – для лингвистического значения “ненормально” (см. рис. 4а).

В случае нескольких экспертов более объективная оценка может быть получена усреднением оценок, данных каждым из экспертов по формуле

$$\bar{\mu}_{\text{ср}}(x_i) = \frac{1}{m} \sum_{j=1}^m a_j(x_i),$$

где m – число экспертов, а величина $a_j(x_i)$ – значение оценки j -м экспертом принадлежности элемента x_i данному лингвистическому значению (элементу терм-множества). Путем усреднения оценок нескольких экспертов получается, как правило, субнормальное нечеткое множество (исключением будет случай, если единичные оценки в какой-нибудь точке будут даны всеми без исключения экспертами). В качестве примера приведем значения функции принадлежности, полученные усреднением оценок нескольких экспертов для лингвистической переменной “артериальное давление”, обсуждаемый универсум которой составляет от 70 до 170 мм. рт. ст.

$$\begin{aligned} \mu(70) &= 0.023, \quad \mu(80) = 0.032, \quad \mu(90) = 0.054, \\ \mu(100) &= 0.09, \quad \mu(110) = 0.157, \quad \mu(120) = 0.28, \\ \mu(130) &= 0.157, \quad \mu(140) = 0.09, \quad \mu(150) = 0.054, \\ \mu(160) &= 0.032, \quad \mu(170) = 0.023. \end{aligned}$$

Поделив все величины на максимальное значение функции принадлежности 0.28, пронормируем это множество и получим следующие значения

Таблица 1

Интенсивность важности	Качественная оценка	Объяснения
0	Несравнимость	Нет смысла сравнивать элементы
1	Одинаковая значимость	Элементы равны по значимости
3	Слабое предпочтение значимости	Существуют не убедительные показания о предпочтении одного элемента над другим
5	Существенное предпочтение значимости	Существует хорошее доказательство того, что один элемент лучше другого
7	Очевидное предпочтение значимости	Существует убедительное доказательство предпочтительности
9	Абсолютное предпочтение значимости	Максимально подтверждается предпочтение одного элемента над другим
2, 4, 6, 8	Промежуточные значения	Необходим компромисс

функции принадлежности:

$$\begin{aligned} \mu(70) &= 0.08, \mu(80) = 0.11, \mu(90) = 0.19, \\ \mu(100) &= 0.28, \mu(110) = 0.56, \mu(120) = 1.0, \\ \mu(130) &= 0.56, \mu(140) = 0.28, \mu(150) = 0.19, \\ \mu(160) &= 0.11, \mu(170) = 0.08. \end{aligned}$$

На рис. 4б приведен график лингвистической переменной “артериальное давление” с лингвистическими значениями “нормально” и “ненормально”.

Рассмотренные прямые методы построения функции принадлежности обладают существенным недостатком – субъективизмом сформулированных оценок. Косвенные методы позволяют получить более объективные оценки. Наиболее известным из них является метод попарных сравнений, предложенный в [8].

По этому методу функции принадлежности определяются по матрице попарных сравнений $M = |m_{ij}|$, элементы которой m_{ij} представляют собой оценки интенсивности принадлежности элементов строки $x_i \in X$ нечеткому множеству по сравнению с элементом столбца $x_j \in X$ (X – обсуждаемый универсум). Таким образом матрица попарных сравнений M становится матрицей отношений, где $m_{ij} = r_i/r_j$. Значения интенсивности r_i, r_j выбираются из таблицы, интерпретирующей относительную интенсивность значений x_i по сравнению с x_j (см. табл. 1).

В табл. 2 дан пример заполнения матрицы попарных сравнений для лингвистической переменной “содержание гемоглобина” (терм “нормально”). Суммируя значения в табл. 2 по столбцам, определим вектор

$$r = (0.022; 0.023; 0.024; 0.026; 0.033; 0.17; 0.24; 0.25; 0.26).$$

Разделив компоненты на максимальное значение 0.26, получим пронормированные значения функ-

ции принадлежности

$$\begin{aligned} \mu(91) &= 0.08, \mu(92) = 0.09, \mu(93) = 0.09, \\ \mu(94) &= 0.1, \mu(95) = 0.15, \mu(96) = 0.65, \\ \mu(97) &= 0.92, \mu(98) = 0.96, \mu(99) = 1, \end{aligned}$$

график которой проведен на рис. 4в. Аналогичным образом построен график зависимости $\mu(x)$ от частоты собственного дыхания, представленный на рис. 4г.

Для получения всех необходимых для синтеза контроллера исходных данных нам осталось рассмотреть выходную лингвистическую переменную y – давление газовой смеси, подаваемой пациенту.

Пусть терм-множество для лингвистической переменной “давление газовой смеси” представлено тремя лингвистическими значениями: повысить немного, повысить сильно, повысить максимально. Результаты обработки экспертных оценок функции принадлежности этим значениям по описанной методике приведены на рис. 5.

Анализируя результаты обработки экспертных оценок, приведенные на рис. 4, 5, нетрудно увидеть, что в соответствии с общим описанием

Таблица 2

%	91	92	93	94	95	96	97	98	99
91	1	1	1	1	1/5	1/7	1/9	1/9	1/9
92	1	1	1	1	1/4	1/7	1/9	1/9	1/9
93	1	1	1	1	1/3	1/7	1/9	1/9	1/9
94	1	1	1	1	1/2	1/5	1/9	1/9	1/9
95	5	4	3	2	1	1/5	1/7	1/7	1/9
96	7	7	7	5	5	1	2	1/3	1/3
97	9	9	9	9	7	2	1	1	1
98	9	9	9	9	7	3	1	1	1
99	9	9	9	9	9	3	1	1	1

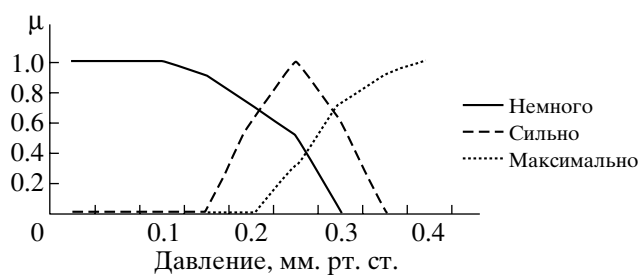


Рис. 5. График зависимости функции принадлежности $\mu(x)$ термам: “немного”, “сильно”, “максимально” от давления газовой смеси, подаваемой пациенту.

работы контроллера нечеткий вывод в данной постановке задачи управления не требуется. На каждом графике точки пересечения функций принадлежности (аналогично рис. 2) делят обсуждаемый универсум на зоны реагирования. Например, для последнего графика получаем отрезки: $(0.31 < x \leq 0.4)$ – максимально; $(0.22 < x \leq 0.31)$ – сильно; $(0.1 < x \leq 0.22)$ – немного.

Выход контроллера (управляемый параметр) определяется ситуациями, которые складываются на его входах.

5. База продукционных правил. В общем случае база продукционных правил содержит правила вида $A_i \rightarrow B_j$. В нашем случае A_i – возможные ситуации, а B_j – предпринимаемые действия. Множество возможных ситуаций определяется врачом-экспертом так же, как и действия, которые необходимо предпринять в каждой ситуации.

Таблица 3

y_1	y_2	Действие
0	0	Снизить давление
0	1	Поднять немного
1	0	Поднять сильно
1	1	Поднять максимально

Таблица 4

s	x_1	x_2	x_3	x_4	y_1	y_2
s_1	1	1	1	1	0	0
s_2	1	1	1	0	0	1
s_3	1	0	1	1	0	1
s_4	0	1	1	1	0	1
s_5	0	0	1	1	1	0
s_6	1	0	1	0	1	0
s_7	1	1	0	1	1	1
s_8	1	1	0	0	1	1
s_9	1	0	0	1	1	1

Для формализации требуемой базы правил воспользуемся специальным видом продукционных правил, а именно логическими продукционными правилами, удобными для сильно недоопределенных автоматных моделей [2]. Эти правила имеют вид $\phi \vdash \psi$, где ϕ и ψ – булевы функции. Введенные в разд. 4 переменные x_1, \dots, x_4 в соответствии с графиками на рис. 4 принимают только два значения: “нормально” (x_i) и “ненормально” (\bar{x}_i).

Конъюнкцию переменных $\tilde{x}_1 \tilde{x}_2 \tilde{x}_3 \tilde{x}_4$, где $\tilde{x}_i \in \{x_i, \bar{x}_i\}$, будем называть текущей ситуацией. По условиям работы контроллера каждой текущей ситуации необходимо поставить в соответствие вполне определенное (врачом-экспертом) действие. Если выполнение действий вынуждают несколько ситуаций, то они могут быть объединены в одном правиле знаком дизъюнкции, что будет означать, что действие предпринимается под влиянием той или (дизъюнкция) иной ситуации.

Пусть, например, ситуация, когда все четыре входных параметра находятся в норме, в соответствии со сформулированными условиями работы контроллера позволяет снизить давление подаваемой пациенту газовой смеси. Это можно представить в виде следующего логического продукционного правила (слева – условие, справа – действие):

$$x_1 x_2 x_3 x_4 \vdash \text{снизить давление}$$

(знак конъюнкции опущен).

Действию “давление поднять немного” по информации врачей-экспертов могут соответствовать следующие три исходные ситуации, что можно отразить в виде следующего правила:

$$\bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee x_1 x_2 x_3 \bar{x}_4 \vdash \text{поднять немного.}$$

Смысл правила в том, что переход любого входного параметра, кроме x_3 (“содержание гемоглобина”) из состояния “норма” в состояние “не норма” при условии сохранения остальных входных параметров в состоянии “норма” приводит к необходимости отказаться от снижения давления подаваемой газовой смеси и снова немного поднять его. Выход за пределы состояния “норма” содержания гемоглобина приводит, по свидетельствам врачей-экспертов, к необходимости более серьезных действий: “поднять давление максимально”, так же, как и в некоторых других случаях, определяемых следующим правилом:

$$x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vdash \text{поднять максимально.}$$

Заметим, что это правило может быть представлено в более компактном виде благодаря очевидным упрощениям в левой его части. Первая и вторая конъюнкции, а также первая и тре-

тя отличаются значением одной переменной, и следовательно, могут быть “склеены” [2]. Тогда правило примет вид

$$x_1x_2\bar{x}_3 \vee x_1\bar{x}_3x_4 \text{ — поднять максимально.}$$

Наконец, к действию “поднять сильно”, по мнению врачей-экспертов, могут привести две ситуации, отраженные в последнем правиле

$$\bar{x}_1\bar{x}_2x_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \text{ — поднять сильно.}$$

Итак, работа входного блока проектируемого контроллера может быть описана всего четырьмя правилами в соответствии с четырьмя возможными действиями. Закодировав четыре действия в соответствии с (1.1) с помощью двух кодовых переменных y_1 и y_2 (см. табл. 3), можно приступить к выполнению шага 8 из разд. 3.

Автоматная таблица, описывающая работу логического контроллера, строится на основании приведенных выше логических продукционных правил и кодов из табл. 3. Таблица 4 является сильно недоопределенной таблицей истинности для комбинационной схемы с четырьмя входами и двумя выходами (из 16 возможных наборов используются всего 9). С помощью матриц Карно [9] нетрудно получить минимальные выражения для двух булевых функций y_1 и y_2 (см. табл. 5 и 6):

$$y_1 = \bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_2\bar{x}_4; \tag{5.1}$$

$$y_2 = \bar{x}_3 \vee \bar{x}_1x_2 \vee x_2\bar{x}_4 \vee x_1\bar{x}_2x_4. \tag{5.2}$$

Синтез комбинационной схемы, реализующей эти функции, оставляем читателю.

6. Реализация схемы контроллера. Вернемся к вопросам организации многоуровневых структур интеллектуальных управляющих систем, затронутым во введении. В [9] предложена методология построения многоуровневых структур интеллектуальных систем управления, в соответствии с которой в системах такого класса выделяются три слоя уровней: организационный, координационный и исполнительный. Организационный слой предназначен для анализа внешних ситуаций, их классификации и принятия решений. Уровни координационного слоя предназначены для разработки и реализации плана решений по управлению с формированием управляющих воздействий, передаваемых на исполнительный уровень. Применительно к нашей задаче синтеза структуры логического контроллера следует заметить, что проблемы анализа и классификации внешних ситуаций уже решены в процессе подготовки исходных данных для проектирования структуры контроллера и реализуются с использованием двух блоков: аналогово-цифрового преобразователя и комбинационной схемы. На выходах последней получаем четыре класса ситуаций, каждому из которых соответствует единственное

Таблица 5. Выходная функция y_1

x_3	x_4	x_1x_2							
		0	0	0	1	1	1	1	0
0	0	–	–	–	–	1	–	–	–
0	1	–	–	–	–	1	–	–	1
1	1	1	–	–	0	0	–	–	0
1	0	–	–	–	–	0	–	–	1

Таблица 6. Выходная функция y_2

x_3	x_4	x_1x_2							
		0	0	0	1	1	1	1	0
0	0	–	–	–	–	1	–	–	–
0	1	–	–	–	–	1	–	–	1
1	1	0	–	–	1	0	–	–	1
1	0	–	–	–	–	1	–	–	0

решение из четырех возможных: “снизить давление”, “поднять немного”, “поднять сильно”, “поднять максимально”. Конкретный план действий в соответствии с возможными решениями (также разработанный заранее) реализуется в блоке “Программа перевода”. Входами для этого блока являются выходы комбинационной схемы (y_1, y_2) и интервалы времени τ , задаваемые врачом. Собственно программа перевода пациента на самостоятельное дыхание здесь не приводится. Напомним ее суть: в исходный момент времени давление дыхательной смеси полностью задается аппаратом ИВЛ, далее давление, подаваемое пациенту, понижается и после фиксированного промежутка времени принимается решение о дальнейшем режиме работы аппарата. Таким образом, существо этой программы состоит в выдерживании предписанного режима вентиляции в течение задаваемого врачом временного интервала τ и отслеживании реакций пациента (выходов комбинационной схемы). Если параметры дыхания в норме ($y_1 = 0, y_2 = 0$), то по истечении временного интервала τ выполняется повторное снижение давления и т.д., вплоть до полного отключения аппарата. Если же ситуация ухудшается, обеспечивается возврат к предыдущему режиму в соответствии со значениями выходов комбинационной схемы.

Окончательно структура логического контроллера представлена на рис. 6. Как видно из рис. 6, информация о реакции пациента подается от датчиков на аналогово-цифровые преобразователи (АЦП) на входе контроллера. Далее преобразованные сигналы поступают в комбинационную схему (КС), выходы которой связаны с блоком “Программа перевода”. Реализуемая этим бло-

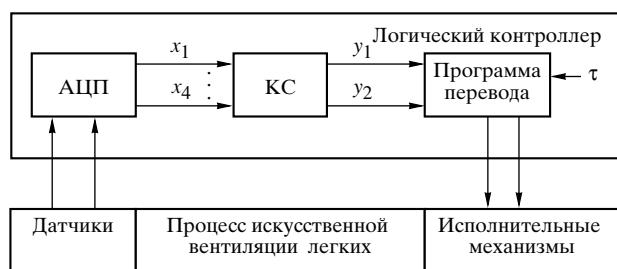


Рис. 6. Контроллер перевода пациента на самостоятельное дыхание.

ком программа будет уточняться и совершенствоваться по результатам апробации методики перевода пациента с ИВЛ на самостоятельное дыхание.

В заключение подчеркнем важный момент, относящийся к особенностям работы логического контроллера. Отмечавшаяся выше аналогия с работой организационного и координационного уровней интеллектуальных управляющих систем не позволяет относить спроектированный контроллер к указанному классу. Дело в том, что задачи анализа и классификации ситуаций, а также составления плана решений по управлению в данном случае реализуются не в процессе функционирования контроллера, а решены заранее с использованием механизмов обработки нечетких знаний и как бы “зашиты” в структуру контроллера. Поскольку при работе контроллера (в процессе формирования управляющих воздействий) механизмы обработки знаний не используются, данный контроллер следует относить к обычным контроллерам, работающим по “жесткой” логике. К интеллектуальным управляющим системам он отношения не имеет.

Заключение. В работе рассмотрены вопросы использования современной нечеткой информационной технологии при разработке обычных ло-

гических контроллеров. При всей простоте и наглядности процессов обработки нечеткой информации блоком КСОНИ реализовать основные процедуры нечеткого вывода и дефазификации не всегда просто и легко, особенно в случае попытки реализовать их аппаратным, а не программным путем. Поэтому, проектируя нечеткие системы управления, полезно поразмышлять, нельзя ли ту же задачу решить с помощью обычной логики. Как показали приведенные исследования, в некоторых случаях это оказывается возможным.

СПИСОК ЛИТЕРАТУРЫ

1. Захаров В.Н. Интеллектуальные системы управления. Основные понятия и определения // Изв. РАН. ТиСУ. 1997. № 3.
2. Захаров В.Н. Интеллектуальные системы логического управления: преодоление неопределенности на исполнительном уровне // Изв. РАН. ТиСУ. 1999. № 5.
3. Справочник по искусственному интеллекту. Т. 2 / Под ред. Д.А. Поспелова. М.: Радио и связь. 1990.
4. Аверкин А.Н., Головина Е.Ю., Сергиевский А.Е. Проектирование нечетких регуляторов на основе треугольных норм // Изв. РАН. ТиСУ. 1997. № 5.
5. Нечеткие множества в задачах управления и искусственного интеллекта / Под ред. Д.А. Поспелова. М.: Наука, 1986.
6. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976.
7. Захаров В.Н., Поспелов Д.А., Хазацкий В.Е. Системы управления. Задание. Проектирование. Реализация. М.: Энергия, 1976.
8. Saaty T.L. Measuring the fuzziness of sets // J. Cybernetics. 1974. V. 4.
9. Saridis G.N. Analytical formulation of the principle of increasing precision with decreasing intelligence for intelligent machines // Automatica. 1989. V. 33. № 2.