

Применение нечеткой логики в имитационной системе автономного адаптивного управления

А.А. Жданов, М.В. Караваев

Аннотация. Работа посвящена исследованию возможностей объединения в одной управляющей системе двух технологий управления – нечеткой логики и метода «автономного адаптивного управления». Обладая своими достоинствами и ограничениями, оба метода при объединении дают новое качество, наделяя систему управления новыми свойствами: 1) системы нечеткой логики оказываются удобными для представления в адаптивной системе управления априорной информации о свойствах объекта управления. Это позволяет в процессе работы системы существенно сократить фазу обучения и быстрее достигать высокого качества управления; 2) использование подсистемы нечеткой логики в адаптивной системе управления повышает как качество управления, так и эффективность метода ААУ в целом. Основным результатом данной работы состоит в том, что впервые показана возможность использования нечеткой логики для адаптивного управления, т.е. для управления, автоматически изменяющего свои свойства непосредственно в процессе управления, в то время как обычным использованием нечеткой логики является построение систем управления на основе знаний эксперта.

1. Введение

В рамках направления, называемого системами «искусственного интеллекта» (ИИ), был разработан ряд известных методов, позволяющих решать различные задачи управления без использования математических моделей управляемых объектов. Именно, возможность обходиться без сложной и дорогостоящей разработки математических моделей составляет основное достоинство этих методов. Это достоинство тем более существенно, что для подавляющего большинства как искусственных, так и естественных объектов, которыми нам хотелось бы управлять, построение точных математических моделей практически невозможно, такие объекты называют плохо формализуемыми. Ситуация еще усложняется, если свойства объектов изменяются в процессе управления.

Сам термин «искусственный интеллект» в последнее время не пользуется популярностью. Причин тому несколько: прежде всего, так и не определилось

однозначно само понятие «интеллекта», который интуитивно ассоциируется со свойствами человеческого мозга; слишком очевидны отличия любой системы ИИ от свойств мозга; принципы построения систем ИИ крайне редко связываются с принципами работы мозга, а сами эти принципы еще не поддаются пониманию. Однако свойства систем ИИ позволяют применять их для решения прикладных задач, самая распространенная из которых – построение систем управления бытовой электроникой, узлами автомобилей, промышленными роботами. Особенно преуспели в этом направлении японские корпорации Matsushita, Hitachi, Mitsubishi, Nissan. Большинство этих разработок носят закрытый характер. К тому же они обычно не претендуют на решение фундаментальных проблем в области ИИ. Более того, можно видеть, что отдельные системы ИИ имитируют лишь разные подзадачи, решаемые естественными нервными системами или мозгом, и не способны к решению этих подзадач всех вместе. Тенденции объединения различных направлений ИИ начинают появляться в разработках последних лет. Эти тенденции естественным образом возникают в каждом из направлений ИИ тогда, когда их разработчики пытаются анализировать работу нервных систем и приближать к ним свои системы. Тем самым, нервные системы являются «центром притяжения» всех направлений ИИ. Поэтому объединение идей различных систем ИИ является естественным и в перспективе ведет к получению не только систем ИИ, имеющих новые свойства, но и к решению фундаментальных проблем ИИ.

В настоящей работе представлены результаты, полученные авторами при объединении двух технологий ИИ. Одна из этих технологий хорошо известна, это системы нечеткой логики, другая технология – это метод автономного адаптивного управления (ААУ), развиваемый в ИСП РАН.

Метод ААУ [1-4] разработан на основе развиваемой авторами концептуальной модели нервной системы. Управляющая система (УС), построенная по методу ААУ, представляет собой единый самообучающийся распознающий-управляющий комплекс. Управляющая система ААУ строится с учетом следующих четырех исходных условий:

1. Условие дискретности, означающее, что система строится из конечного числа элементов (нейронов, связей между ними, блоков памяти и т.д.);
2. Условие автономности, означающее, что УС является подсистемой ОУ и получает знания только на основании своего эмпирического опыта;
3. Условие минимума исходных знаний означает, что на момент начала процесса управления УС располагает минимумом знаний о свойствах объекта и среды его обитания;
4. Условие максимальной начальной приспособленности говорит о возможности системы управления обеспечивать минимальное качество управления на начальном этапе жизнедеятельности объекта управления.

В УС взаимно согласованно решаются несколько задач, именно: задача

автоматической кластеризации (задача формирования образов), задача распознавания образов, задача получения знаний о свойствах объекта управления на основе эмпирического опыта (задача поиска закономерностей в предыстории системы), представление знаний, задача принятия решений и некоторые другие. Для решения этих задач в УС предусмотрены соответствующие подсистемы. Роль «учителя» в самообучаемой управляющей системе, построенной по методу ААУ, играет подсистема, названная «аппаратом эмоций». Эта подсистема выполняет в УС несколько функций, в частности, она вырабатывает некоторый функционал, который УС пытается оптимизировать. Текущее значение этого функционала должно по возможности объективно отражать соответствие состояния объекта управления и заданных целевых функций. Множество целевых функций образует определенную иерархическую систему, вершиной которой являются две цели – «выживание объекта управления» и «накопление знаний управляющей системой».

Принцип работы УС ААУ в целом состоит в следующем. Объект управления (ОУ) несет на себе множество датчиков, регистрирующих параметры, обеспечивающие наблюдаемость и управляемость ОУ, а также состояние исполнителей. В потоке входной информации, поступающей от датчиков, автоматически обнаруживаются пространственно-временные регулярности определенного вида – *образы*. Те образы, прообразы которых включают в себя причинно-следственные связи состояний ОУ и внешних воздействий УС, являются эмпирическими *знаниями*, отражающими функциональные свойства ОУ. Эти знания сохраняются в разделе памяти УС, названном *базой знаний*. Статистическими способами находятся зависимости различных состояний ОУ и упомянутого выше функционала, отражающего состояние ОУ. При принятии решений УС распознает образы, описывающие текущее состояние ОУ, обращается к базе знаний и находит в ней такие действия, которые максимизируют функционал. Исполнителям дается команда на совершение принятого решения. Адаптивные свойства УС ААУ состоят в том, что в реальном времени управления а) формируются новые образы, в том числе – образы над образами (образы большего порядка); б) для новых образов находятся их оценки качества, что тем самым расширяет размерность максимизируемого функционала; в) находятся новые знания; г) происходит агрегирование знаний при обнаружении знаний для образов большего порядка. Эти подзадачи решаются подсистемами УС, каждая из которых использует некоторый метод автоматического поиска закономерностей, например, статистических закономерностей.

Одна из основных проблем метода ААУ состоит в том, что для подсистем УС, реализующих тот или иной автоматический метод поиска закономерности в своих входных данных, необходимо заранее указать классы таких закономерностей, либо правила, по которым эти классы следует формировать, т.е. задать гипотезы, которые будет проверять УС. Известно, что задание априорной информации требуется для любой самообучаемой системы

распознавания. Причина этого состоит в том, что для описания всякой входной информации можно использовать огромное число различных базисов. Также можно искать закономерности в заданных данных в многочисленных базисах. Построить такую распознающую систему, которая проверяла бы все мыслимые гипотезы во всех мыслимых базисах, невозможно. Подобно тому, как всякий живой организм приспособлен к работе в определенном для него ограниченном базисе знаний, УС ААУ требует априорного задания такого базиса. В живой природе такой базис находится и уточняется в процессе эволюционного отбора. В системе ААУ требуется разработка специальной методики представления в БЗ имеющейся априорной информации об ОУ, среде и т.д.

Одну из возможностей представления априорной информации в УС ААУ мы увидели в нечеткой логике (fuzzy logic), в последнее время часто используемой для построения управляющих систем. Сегодня нечеткая логика применяется лишь для формальной записи знаний эксперта, что позволяет использовать их для автоматического принятия решений, аналогичных решениям эксперта. Мы не нашли в литературе упоминаний об управляющих системах на основе нечеткой логики, которые допускали бы самообучение в процессе управления. Управляющая система, которую мы представляем в данной работе, является самообучаемой. Таким образом, мы предлагаем способ использования нечеткой логики для задания априорной информации в самообучаемой (адаптивной) системе управления.

2. Основы теории нечеткой логики

Нечеткая логика (НЛ) появилась в 1965 в работах [5,6] Лотфи А. Заде, профессора технических наук Калифорнийского университета в Беркли.

Нечеткая логика является многозначной логикой. В отличие от традиционной математики, требующей на каждом шаге моделирования точных и однозначных формулировок закономерностей, нечеткая логика предлагает совершенно иной уровень мышления, при котором творческий процесс моделирования происходит на наивысшем уровне абстракции, и постулируется лишь минимальный набор закономерностей.

Значения нечетких переменных, получаемые в результате неточных измерений, во многом аналогичны распределениям теории вероятностей, но свободны от присущих последним недостатков. К таким недостаткам относятся: малое количество пригодных к анализу функций распределения, необходимость их принудительной нормализации, соблюдение требований аддитивности, а также трудность обоснования адекватности математической абстракции для описания поведения фактических величин. В пределе, при возрастании точности, нечеткая логика переходит в Булеву алгебру. По сравнению с вероятностным методом, нечеткий метод позволяет резко сократить объем производимых вычислений, что приводит к увеличению быстродействия нечетких систем.

Нечеткая логика первоначально использовалась как наиболее удобный способ построения систем управления метрополитенами и сложными технологическими процессами, в дальнейшем она нашла применение в бытовой электронике, различных системах управления, диагностических и других экспертных системах.

2.1 Нечеткие множества и основные операции над ними

Пусть E - универсальное множество, x - элемент E . Нечеткое подмножество A универсального множества E определяется как множество упорядоченных пар $A = \{\mu_A(x)/x\}$, где $\mu_A(x)$ - характеристическая функция принадлежности (или просто функция принадлежности), принимающая значения в некотором вполне упорядоченном множестве M (например, $M = [0,1]$). Функция принадлежности указывает степень (или уровень) принадлежности элемента x подмножеству A . Пример графического представления функции принадлежности представлен на рис. 1. Множество M иногда называют множеством принадлежностей. Если $M = \{0,1\}$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество.

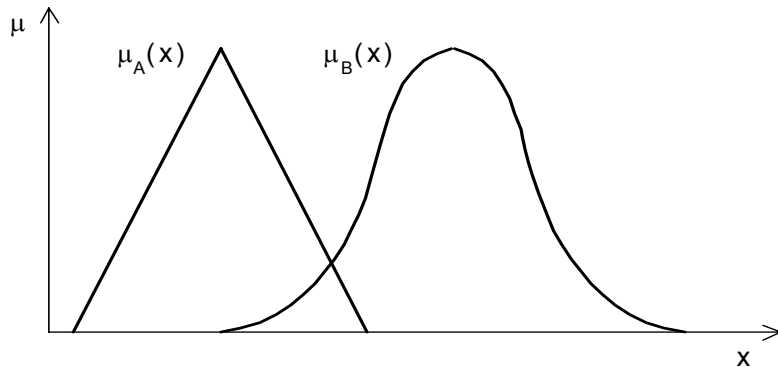


Рис. 1. Примеры функций принадлежности.

Над нечеткими множествами определены следующие основные операции.

1. Дополнение: пусть $M=[0,1]$, A и B – нечеткие множества, заданные на E . A и B дополняют друг друга, если $\forall x \in E \mu_A(x) = 1 - \mu_B(x)$. Обозначается “ $A = \hat{B}$ ” или “ $B = \hat{A}$ ”.
2. Пересечение: $A \cap B$ – наибольшее нечеткое подмножество, содержащееся одновременно в A и B . $\forall x \in E \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$.
3. Объединение: $A \cup B$ – наименьшее нечеткое подмножество, включающее как A , так и B с функцией принадлежности $\forall x \in E \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$.
4. Разность: $A - B = A \cap \hat{B}$ с функцией принадлежности $\forall x \in E \mu_{A-B}(x) = \mu_{A \cap \hat{B}}(x) = \min(\mu_A(x), 1 - \mu_B(x))$.

5. Дизъюнктивная сумма: $A \oplus B = (A - B) \cup (B - A) = (A \cap \hat{B}) \cup (\hat{A} \cap B)$ с функцией принадлежности $\forall x \in E \mu_{A \oplus B}(x) = \max\{\min(\mu_A(x), 1 - \mu_B(x)), \min(1 - \mu_A(x), \mu_B(x))\}$.
6. Алгебраическое произведение: $A \cdot B$ с функцией принадлежности $\forall x \in E \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x)$.
7. Алгебраическая сумма: $A + B$ с функцией принадлежности $\forall x \in E \mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$.
8. Возведение в степень: A^α с функцией принадлежности $\forall x \in E \mu_{A^\alpha}(x) = \mu_A^\alpha(x)$.
Частные случаи возведения в степень: $CON(A) = A^2$ - операция концентрирования, $DIL(A) = A^{0.5}$ - операция растяжения.

2.2 Нечеткие и лингвистические переменные. Общий алгоритм нечеткого управления

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, A \rangle$, где α – наименование переменной, X – универсальное множество (область определения α), A – нечеткое множество на X , описывающее ограничения (т.е. $\mu_A(x)$) на значения нечеткой переменной α .

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где

- β – наименование лингвистической переменной;
- T – множество ее значений (терм-множество), представляющих собой наименования нечетких переменных, область определения каждой из которых является множеством X . Множество T называется базовым терм-множеством лингвистической переменной;
- G – синтаксическая процедура, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы (значения). Множество $T \cup G(T)$, где $G(T)$ – множество сгенерированных термов, называется расширенным терм-множеством лингвистической переменной;
- M – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, то есть сформировать соответствующее нечеткое множество.

Примером лингвистической переменной может служить слово “вода”, которая включает в себя следующие нечеткие переменные (множество значений): “холодная”, “теплая” и “горячая”.

Из лингвистических переменных строятся нечеткие высказывания. Они часто представляют собой предложения естественного языка, что позволяет, например, использовать знания экспертов для построения систем управления на основе нечеткой логики.

Подмножество нечетких высказываний, используемое в системах нечеткого управления, называется правилами, которые обычно представляют собой

выражения вида “Если ... то ...”. В левой части правил находятся входные лингвистические переменные, сопоставленные с их возможными значениями, а в правой – выходные. В правилах нечеткие переменные могут объединяться при помощи связок “и”, “или” и “не”, которые соответствуют операциям пересечения, объединения и дополнения над нечеткими множествами.

Например, простейший набор правил для управления температурой и давлением воды в смесителе может выглядеть следующим образом:

1. Если вода горячая и давление слабое, то повернуть синий вентиль вправо;
 2. Если вода горячая и давление сильное, то повернуть красный вентиль влево;
 3. Если вода теплая, то не вращать вентили;
 4. Если вода холодная и давление слабое, то повернуть красный вентиль вправо;
 5. Если вода холодная и давление сильное, то повернуть синий вентиль влево;
- где “вода” и “давление” – лингвистические переменные, а “холодная”, “теплая”, “горячая” и “слабое”, “сильное” – их значения – нечеткие переменные, задаваемые нечеткими множествами.

Процесс преобразования входных физических величин в значения лингвистических переменных при помощи функций принадлежности называется фаззификацией.

Например, сопоставление физического значения температуры воды с множествами значений лингвистической переменной “вода” можно произвести следующим образом (рис. 2): степень принадлежности физического значения температуры к нечеткой переменной (нечеткому множеству) “холодная” равна 0,10, к нечеткой переменной “теплая” – 0,65, а к переменной “горячая” – 0,00.

При обработке нечетких правил применяются стандартные операции над нечеткими множествами, в результате чего происходит вычисление степеней принадлежности выходных физических воздействий к значениям лингвистических переменных.

Если для рассмотренного выше примера из пяти правил принять значения лингвистической переменной “давление”: “слабое” – 0,35, а “сильное” – 0,78, то результаты работы правил будут выглядеть следующим образом:

1. Повернуть синий вентиль вправо: $\min(0,00; 0,35) = 0,00$;
2. Повернуть красный вентиль влево: $\min(0,00; 0,78) = 0,00$;
3. Не вращать вентили: $0,65 = 0,65$;
4. Повернуть красный вентиль вправо: $\min(0,10; 0,35) = 0,10$;
5. Повернуть синий вентиль влево: $\min(0,10; 0,78) = 0,10$.

Процесс вычисления значений выходных физических переменных называется дефаззификацией и происходит обычно в соответствии с формулами нахождения средневзвешенной величины. Дефаззификация так же часто производится нахождением центра масс фигуры, образованной линиями функций принадлежности соответствующих нечетких переменных выходной лингвистической переменной, осью абсцисс и прямыми, параллельными оси

абсцисс и отстоящими от нее на величину степени принадлежности физической переменной соответствующей нечеткой переменной.

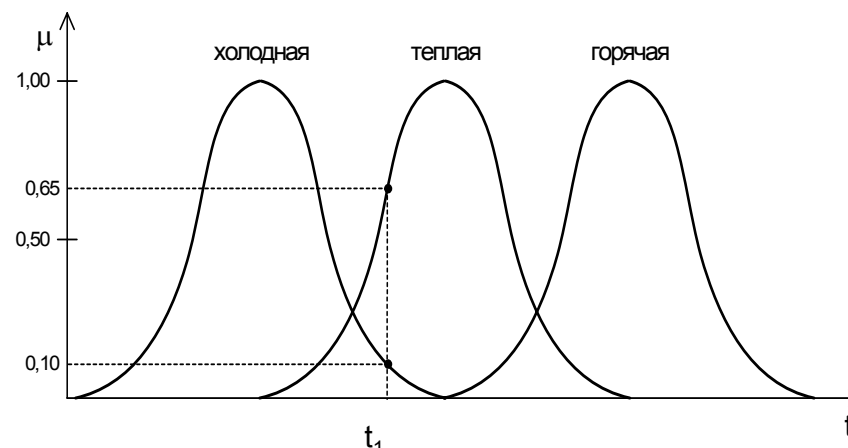


Рис. 2. Сопоставление физического значения температуры t с нечеткими переменными “холодная”, “теплая”, “горячая”.

Для рассмотренного выше примера допустим, что термин “повернуть вправо” означает поворот вентиль на 30 градусов, а “повернуть влево” – на –30 градусов. Тогда значение угла поворота для красного вентиль будет равно: $(0,65 \cdot 0 + 0,1 \cdot 30) / (0,65 + 0,1) = 4$ градуса. Аналогичное значение получится и для синего вентиль.

Таким образом, общий алгоритм нечеткого управления состоит из следующих взаимосвязанных процедур:

1. преобразование входных физических величин, получаемых от измерительных датчиков с объекта управления, в значения лингвистических переменных – фаззификация;
2. обработка нечетких высказываний относительно входных и выходных лингвистических переменных системы;
3. преобразование выходных лингвистических переменных системы в физические управляющие переменные – дефаззификация.

Недостатками нечетких систем являются:

1. отсутствие стандартной методики конструирования нечетких систем;
2. невозможность математического анализа нечетких систем существующими методами;
3. применение нечеткого подхода, в отличие от вероятностного подхода, не приводит к повышению точности вычислений;
4. отсутствие адаптивности в реальном времени.

3. Построение подсистемы начальной адаптации системы ААУ

Рассмотрим возможности применения нечеткой логики в управляющих системах, построенных по методу ААУ.

Для обеспечения одного из основных начальных условий системы, построенной по методу ААУ – максимальной начальной приспособленности, необходим некоторый аппарат, заменяющий действие механизмов естественного отбора и передачи наследственной информации, присущих живым организмам.

Одним из возможных вариантов обеспечения этого свойства в моделируемых нервных системах является введение некой подсистемы с заранее заложенными в нее базовыми правилами управления, основанными на общих знаниях об объекте и предполагаемой среде его функционирования. Эта подсистема должна определять основные принципы управления на начальном этапе жизнедеятельности объекта, когда основная система управления еще не успела накопить достаточно информации о закономерных связях между воздействиями на внешнюю среду и состояниями, в которые объект переходит в результате этих воздействий, необходимой для “осмысленного” управления объектом.

Очевидно, что управление на основе только априорных знаний будет менее качественным, чем управление, осуществляемое системой ААУ с эмпирически заполненной базой знаний. Напомним, что нас интересуют плохо формализуемые объекты управления. Основная цель вводимой подсистемы – обеспечение минимального качества управления объектом на начальном этапе.

В качестве подсистемы нечетких исходных знаний (так был назван новый блок системы ААУ, дополняющий ее априорными знаниями) выбрана система управления, построенная на основе нечеткой логики.

На примере прикладной системы – прототипа системы стабилизации углового движения автоматического космического аппарата “Пилот”, рассмотрим работу подсистемы нечетких исходных знаний. Подробно система “Пилот” описана в [1].

Общая структура системы ААУ с модулем нечетких исходных знаний изображена на рис. 3.

Входными данными для него являются физические значения текущего углового отклонения $f(t)$ и угловой скорости df/dt , которые отображаются в нечеткие множества (переменные) входных лингвистических переменных “угловое отклонение” и “угловая скорость”.

Выходные данные модуля – усредненное значение номера управляющего воздействия, полученное из выходной лингвистической переменной “управляющее воздействие”.

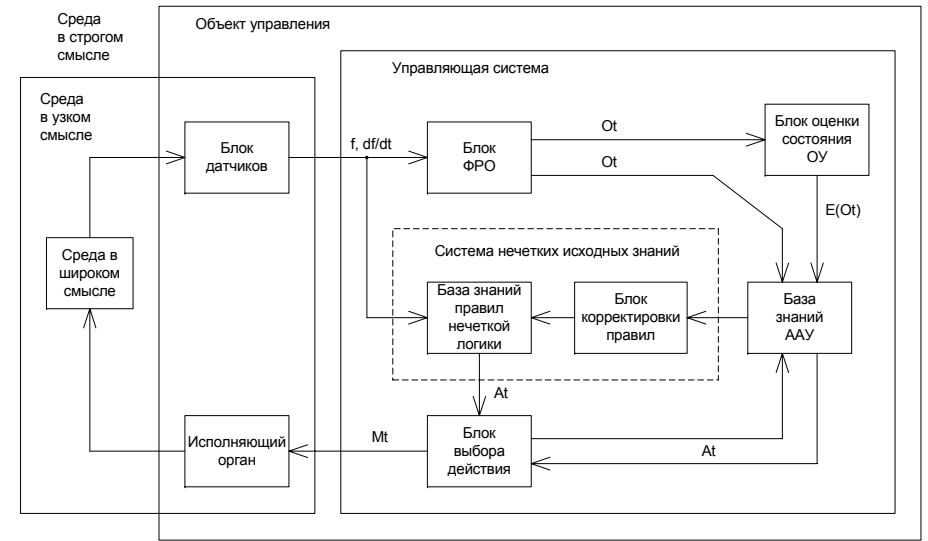


Рис. 3. Структура системы ААУ с подсистемой нечетких исходных знаний.

База знаний модуля состоит из набора правил, заданных при помощи нечетких переменных $F_1...F_N$, $dF_1...dF_M$, и $A_1...A_{N*M}$, минуя уровень лингвистических переменных “угловое отклонение”, “угловая скорость” и “выходное воздействие”:

$$\begin{aligned}
 &F_1 \& dF_1 \rightarrow A_1 \\
 &F_1 \& dF_2 \rightarrow A_2 \\
 &\dots \dots \dots \\
 &F_1 \& dF_M \rightarrow A_M \\
 &F_2 \& dF_1 \rightarrow A_{M+1} \\
 &F_2 \& dF_2 \rightarrow A_{M+2} \\
 &\dots \dots \dots \\
 &F_i \& dF_j \rightarrow A_{(i-1)*M+j} \\
 &\dots \dots \dots \\
 &F_N \& dF_{M-1} \rightarrow A_{N*M-1} \\
 &F_N \& dF_M \rightarrow A_{N*M} ,
 \end{aligned}$$

где N и M – количества нечетких множеств, на которые разбит диапазон наблюдаемых значений угла отклонения и угловой скорости космического аппарата.

Результаты сработавших правил используются для вычисления номера A управляющего воздействия по следующей формуле:

$$A = \frac{\sum (A_k \cdot P_k)}{\sum P_k}, \quad (1) \quad (5)$$

где A_k – номера сработавших правил, P_k – выходные значения соответствующих правил.

Также возможен вариант управления, при котором среди сработавших правил выбирается одно правило с максимальным результатом и выполняется действие, заданное только этим правилом.

В смоделированной системе управления на основе НЛ используются 3 вида функций принадлежности $p(x)$ входной переменной x одному из нечетких множеств наблюдаемых переменных: линейная (треугольная), квадратичная и обратная квадратичной. Здесь i – номер интервала ($i \in [0..N]$), x – безразмерная величина, полученная из значения входной физической переменной по следующей формуле:

$$x = \frac{f + F_{MAX}}{2 \cdot F_{MAX}} \cdot N, \quad (2)$$

где f – значение входной физической переменной, F_{MAX} – максимальное абсолютное значение входной физической переменной (считается, что f может принимать любое значение в диапазоне $[-F_{MAX}; F_{MAX}]$), N – количество нечетких множеств, на которые разбит диапазон наблюдаемых значений физической входной переменной. Причем N может не совпадать с числом интервалов для системы ААУ. Для системы НЛ лучше поделить диапазон наблюдаемых значений на меньшее число более широких интервалов с целью сокращения количества правил в БЗ.

Используемые функции принадлежности:

1. Линейная (треугольная) функция:

$$p_i(x) = \begin{cases} x - i + 1, & i - 1 \leq x < i \\ i - x + 1, & i \leq x < i + 1 \end{cases} \quad (3)$$

2. Квадратичная функция:

$$p_i(x) = 1 - (x - i)^2, \quad i - 1 \leq x < i + 1 \quad (4)$$

3. Функция, обратная квадратичной

$$p_i(x) = \frac{1}{1 + K \cdot (x - i)^2}, \quad i - 1 \leq x < i + 1,$$

где K – априорно задаваемый коэффициент, определяющий площадь графика функции принадлежности (чем больше K , тем меньше площадь).

Рассмотрим общий алгоритм работы такой системы ААУ.

Изначально база знаний системы ААУ пуста, а в базе знаний системы НЛ находятся правила, полученные на основании априорной информации о движении космического аппарата. По этим правилам и производится управление на начальном этапе.

Во время полета космического аппарата база знаний системы ААУ постепенно заполняется и управление плавно переходит к системе ААУ: если ОУ попадает в какое-то состояние, оптимальное управляющее воздействие для которого не содержится в БЗ ААУ (не заполнен столбец БЗ), то право выбора управляющего момента на этом такте передается системе НЛ, а информация об этом состоянии и выбранном системой НЛ воздействии заносится в БЗ системы ААУ. При этом система ААУ получает право выбора действия в каждом состоянии только в том случае, если она может предложить действие, которое переведет ОУ в состояние со значением оценочной функции не ниже некоторого порога E_{min} , задающегося априорно.

При длительном управлении в БЗ системы ААУ накапливаются более достоверные знания, чем те, которые заложены априорно в БЗ нечетких правил, в результате чего возникает необходимость корректировки этих правил системы начальной адаптации для использования их в следующих сеансах работы. Корректировка происходит путем вычисления по БЗ ААУ суммарного выходного управляющего воздействия для каждого нечеткого правила и замещения вычисленной величиной старого значения. При этом непрерывное значение индекса F углового интервала фазовой плоскости системы НЛ получается из соответствующего индекса образа на фазовой плоскости ААУ по следующей формуле:

$$F = \left[\frac{[(X - 1) \cdot 2 - X_{max} + 1]}{X_{max} - 2} + 1 \right] \cdot \frac{F_{max}}{2}, \quad (6)$$

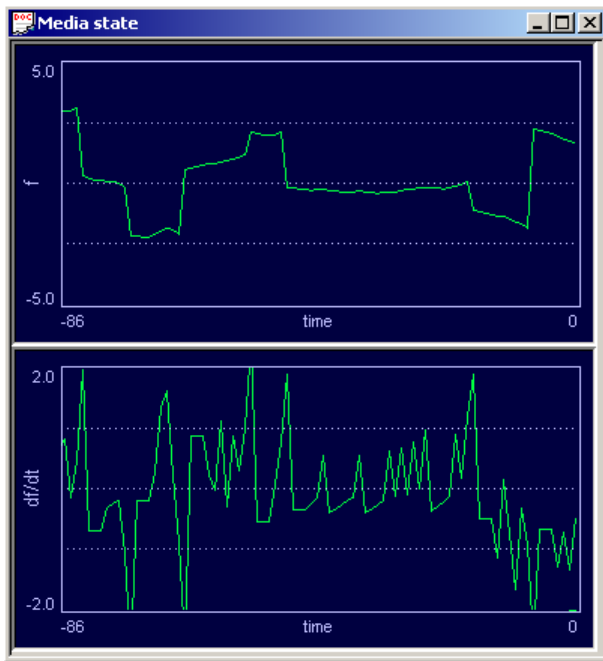
где X – номер углового интервала фазовой плоскости ААУ; X_{max} – количество угловых интервалов фазовой плоскости ААУ (включая открытые); F_{max} – количество нечетких переменных входной лингвистической переменной системы НЛ.

4. Результаты работы программной модели

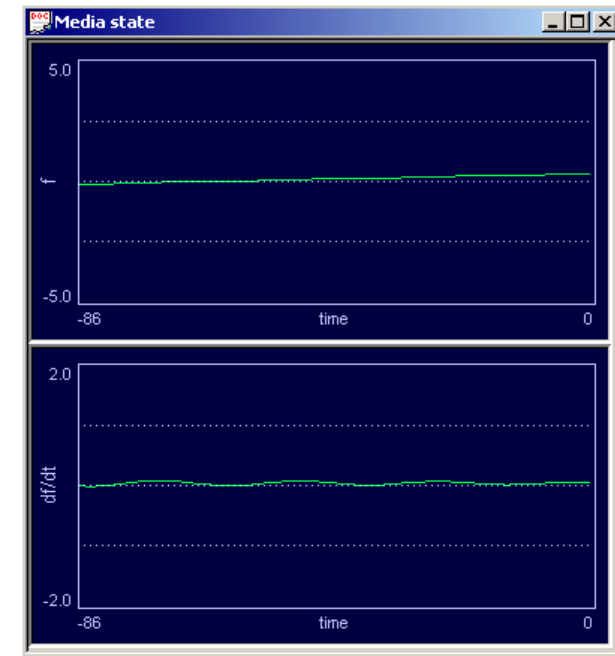
За основу программной модели комплекса взята упомянутая выше программа "Пилот". Добавленный к программе модуль, моделирующий подсистему нечетких исходных знаний, может содержать от 4 до 10000 правил, входные лингвистические переменные которых содержат соответственно от 2 до 100 нечетких переменных.

На рис. 4 приведены два графика, отображающие временные интервалы значений углового отклонения и угловой скорости на начальном этапе управления: с использованием системы нечетких исходных знаний и без нее. По графикам можно видеть, что разработанная подсистема на основе нечеткой логики обеспечивает приемлемое качество управления в начале работы и позволяет избежать полного перебора при самообучении системы ААУ.

Рис. 5 содержит пример заполнения базы знаний нечеткими правилами при разбиении входных лингвистических переменных на 9 нечетких переменных. На пересечении столбца f и строки df находится номер действия, выполняемого при попадании входных физических переменных f и df в нечеткое множество соответствующей входной нечеткой переменной.



а)



б)

Рис. 4. Сравнение качества управления на начальном этапе работы адаптивной системы управления с использованием подсистемы нечетких исходных знаний (б) и без нее (а).

При тестовых запусках программы было определено, что

- а) оптимальным количеством является 25 правил при сопоставлении входных физических переменных f и df 5-ти нечетким переменным;
- б) оптимальной является линейная (треугольная) функция принадлежности входных физических переменных нечетким множествам, к тому же, она наиболее проста для вычисления;
- в) оптимальный вариант объединения выходных результатов сработавших правил – объединение по среднему значению.

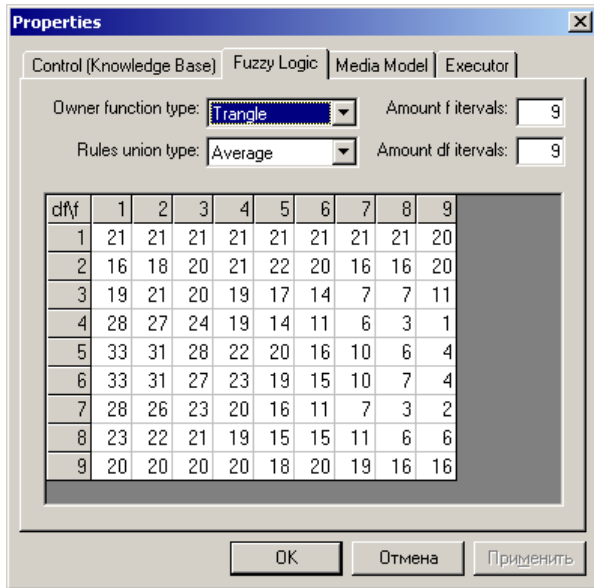


Рис. 5. База знаний, состоящая из 81 нечеткого правила, сформированного по заполненной базе знаний системы ААУ.

При сравнении разработанного варианта управляющей системы ААУ с исходным вариантом (без системы нечетких исходных знаний) были выявлены следующие преимущества:

1. длительность фазы начальной стабилизации объекта управления сократилась в среднем с 7000 до 500 тактов; чем более точно подобраны правила, тем меньше время начальной стабилизации (точность правил ограничивается, естественно, тем, что в нашем случае объект управления плохо формализуем);
2. качество управления повысилось в среднем с 2,0 до 2,9 за первые 2000 тактов и с 2,5 до 3,0 за первые 10000 тактов управления, где количественная оценка, использовавшаяся в программе «Пилот», вычисляется по следующей формуле:

$$\sum_{i=0}^{13} \frac{S_i}{t \cdot 10 + 10} \cdot \frac{i}{2} \quad (7)$$

Здесь t – номер текущего такта, S_i – количество попаданий объекта в состояние с оценкой равной i , за все время работы системы. Оценка $i = 1, 2, \dots, 6$ больше, чем дальше от центра фазовой плоскости находится текущее состояние ОУ.

5. Заключение

В данной работе предложен один из возможных путей совмещения двух технологий управления в одной системе: метода нечеткой логики и метода «автономного адаптивного управления», развиваемого в ИСП РАН. Подсистема нечетких исходных знаний на основе fuzzy logic была встроена в систему управления ААУ, дополняя ее на начальной фазе работы адаптивной системы управления априорными правилами управления, полученными на основе знаний экспертов.

Для испытания разработанной технологии было выбрано одно из практических приложений – программа «Пилот», прототип адаптивной системы управления угловым движением космического аппарата. Этот прототип удобен для сравнения различных вариантов систем управления тем, что наглядно демонстрирует их достоинства и недостатки.

Основные результаты и выводы, полученные в результате компьютерных испытаний разработанной системы заключаются в следующем:

- применение подсистемы нечетких исходных знаний в системе ААУ повышает среднее качество управления за счет ускорения фазы обучения адаптивной системы управления (в прикладном примере высокое качество управления космическим аппаратом достигалось быстрее);
- при заполнении обычной для системы ААУ базы знаний эмпирической информацией блок корректировки нечетких правил формирует адекватные нечеткие правила управления объектом, которые затем могут использоваться экспертами для изучения поведения данного объекта;
- подсистему нечетких исходных знаний нельзя применить к объектам управления, законы управления которыми эксперты не могут сформулировать хотя бы в самом общем, приближенном виде.

6. Направления дальнейших исследований

При дальнейшей разработке данной темы следует рассмотреть также иные возможные пути применения нечеткой логики в системах ААУ, а именно: 1) использовать теорию нечетких множеств в процессе формирования и распознавания образов, заполнения БЗ, и на этапе принятия решений; 2) разработать БЗ, построенную не на основе нейроподобной сети (как в базовой версии), а из нечетких правил, формирующихся в процессе управления.

При использовании рассмотренного пути применения нечеткой логики в системах ААУ с более развитым, нежели у системы «Пилот», аппаратом ФРО (который распознает образы более продолжительных во времени преобразов)

может возникнуть следующая проблема: распознавание образов аппаратом ФРО – дискретный процесс (на выходе дискретные величины: образ распознан или не распознан), а входные переменные системы НЛ должны иметь непрерывный характер. В противном случае БЗ системы НЛ вырождается в набор элементарных продукционных правил, теряя все преимущества нечеткого управления.

При исследовании возможности построения основной БЗ системы ААУ на основе правил нечеткой логики понятие “образ” может претерпеть некоторые изменения, потеряв, в частности, свои дискретные черты: его может заменить множество значений входных физических переменных или их степень принадлежности одной из входных нечетких переменных.

Система управления на основе нечеткой логики также может найти применение в качестве основной управляющей системы, функционирующей за пределами диапазона значений входных переменных, в котором работает система ААУ.

Литература

1. Труды института системного программирования, 1999. Том 1. Под ред. Иванникова В. П. М.: Биоинформсервис, 2000. – 124 с.
2. Zhdanov A. A. About an Autonomous Adaptive Control Methodology. ISIC/CIRA/(ISAS'98), NIST, Gaithersburg, Maryland. September 14-17, 1998, pp. 227-232.
3. Zhdanov A. A., Vinokurov A. N., Emotions Simulation in Methodology of Autonomous Adaptive Control, Proceedings of ISIC'99/ISAS'99, 1999.
4. Информация с сайта <http://www.ispras.ru/~zhdanov/>.
5. Zadeh, Lotfi. Fuzzy Sets / Information and Control, 8(3), June 1965, pp.338-53.
6. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.:Мир, 1976.
7. Пивкин В.Я., Бакулин Е.П., Кореньков Д.И. Нечеткие множества в системах управления, http://idisys.iae.nsk.su/fuzzy_book/.
8. Баер П., Новак С., Винклер Р. Введение в нечеткую логику и системы нечеткого управления, <http://softlab.od.ua/algo/neuro/fuzzy-intro/>