

В.М. ШПАКОВ

**ФОРМАЛИЗАЦИЯ И ИСПОЛЬЗОВАНИЕ НЕЧЕТКИХ ПРЕДИКАТОВ ДЛЯ РЕАЛИЗАЦИИ ГИБРИДНЫХ ПРОЦЕССОВ**

*Шпаков В.М. Формализация и использование нечетких предикатов для реализации гибридных процессов.*

**Аннотация.** Рассматривается методика формализации нечетких предикатов совместно с четкими логическими переменными для спецификации нечетких логико-динамических ситуаций и четких логических действий (дискретных команд). Методика основана на представлении четких и нечетких логических переменных с помощью функций принадлежности и на применении правил нечеткого вывода. При этом использовались только формы представления нечетких логических функций пригодные также и для представления четких логических функций. На примерах показана возможность применения рассматриваемой методики для компьютерной реализации гибридных процессов.

**Ключевые слова:** нечеткая логика, представление знаний, обработка знаний, компьютерное моделирование, автоматическое управление.

*Shpakov V.M. Formalization and Use of Fuzzy Predicates for Hybrid Processes Implementation.*

**Abstract.** The technique of formalization of fuzzy predicates together with crisp logical variables for the specification of fuzzy logic-dynamic situations and crisp logical commands is considered. The technique is based on the submission of crisp and fuzzy logic variables by means of membership functions and on the use of fuzzy inference rules. Here we only used the forms of presentation of fuzzy logic functions which are also suitable for presentation of crisp logic functions. By the examples the possibility of using the considered technique for the computer implementation of hybrid processes is shown.

**Keywords:** fuzzy logic, knowledge representation, knowledge processing, computer simulation, automatic control.

**1. Введение.** Функционирование различных (технических, производственных, химических, биофизических и др.) динамических систем может быть представлено совокупностью взаимодействующих гибридных процессов. Потребность в спецификации и компьютерной реализации таких совокупностей процессов или их частей возникает при моделировании указанных систем и при компьютерной реализации систем управления ими [1]. Наиболее перспективным подходом к спецификации и реализации гибридных процессов является, на наш взгляд, транзитивный ситуационный подход, основанный на использовании модели гибридного автомата [2, 3], функции перехода которого задаются трансформационными правилами [4]. При этом создаются множества вещественных  $X$  и логических  $W$  переменных для представления состояний, соответственно, непрерывных и дискретно-событийных составляющих процессов. Среди логических переменных выделяется подмножество независимых внешних воздействий  $V$ , подмножество переменных для представления состояний дискретных

элементов системы и динамики (режимов) изменения непрерывных состояний  $Q$  и подмножество предикатов от состояний непрерывных процессов, то есть  $W = V \cup Q \cup G$ . Среди вещественных переменных выделяются независимые внешние воздействия  $X_i$  и состояния  $X_s$ .

Трансформационные правила определяют транзитивные отношения между значениями текущих состояний процессов и значениями этих состояний через определенный, достаточно короткий промежуток времени. Они представляют собой формализованные знания о развитии процессов. Для спецификации гибридных процессов необходимо иметь совокупность правил трансформации дискретных (логических) состояний, определяющих функцию перехода  $\sigma : W \rightarrow Q$ , и совокупность правил трансформации непрерывных состояний для функции перехода  $\delta : W \times X \rightarrow X_s$ . Текущее состояние непрерывного процесса зависит как от значений некоторых непрерывных состояний, так и от динамики процесса, определяемой логическими переменными из множества  $W$ . Четкие предикаты от непрерывных состояний задаются с помощью неравенств относительно этих переменных.

Трансформационное правило, определяющее функцию перехода  $\sigma$ , имеет следующий вид [4]:

$$S_j \rightarrow q'_{j_1}, \dots, q'_{j_1}, \dots, q'_{j_m}, \text{ где } q'_{j_i} \in Q \times \{False, True\}, m \in \{1, \dots, |Q|\}, \quad (1)$$

$$S_j = s_{j_1}, \dots, s_{j_1}, \dots, s_{j_m}, s_{j_i} \in W \times \{False, True\}, m \in \{1, \dots, |W|\}. \quad (2)$$

Условная часть  $S_j$  правила (1) представляет собой элементарную конъюнкцию (2) некоторых логических переменных или их отрицаний. Эта конъюнкция в данном контексте интерпретируется как локальная логико-динамическая ситуация. При реализации процессов обрабатывающая процедура интерпретатора правил последовательно обрабатывает списки правил. В случае истинности конъюнкции из условной части правила всем состояниям из его правой части присваиваются логические значения True. Если в текущем цикле обработки правил нет ни одного правила, которое присваивает некоторому логическому состоянию значение True, то этому состоянию присваивается значение False. Для задания функции перехода  $\delta$  используются правила следующего вида:

$$S_j \rightarrow x'_k = \tau_k(x_k, x), x_k \in X_s, x \in X. \quad (3)$$

Это правило в случае истинности ситуации  $S_j$  запускает процедуру, соответствующую заданному транзитивному отношению  $\tau_k(x_k, x)$ , которая на основании текущих значений непрерывных состояний  $x_k$  и  $x$  вычисляет следующее значение состояния  $x_k'$ .

Подход к спецификации и компьютерной реализации гибридных процессов, основанный на использовании трансформационных правил, экспериментально исследуется в СПИИРАН в течение ряда лет. Было показано, что он может быть эффективно использован при компьютерном моделировании больших промышленных установок [5]. Подход применим в случаях, когда исходные знания о развитии непрерывных процессов представлены в виде обыкновенных дифференциальных уравнений, в форме структурных схем и передаточных функций или в виде физических моделей [6]. При этом знания о логике развития процессов должны быть представлены в виде четких логических функций. Однако при моделировании и разработке систем управления возникает необходимость в использовании нечетких знаний. Существуют два направления использования нечетких знаний [7]. Во-первых, на основании нечетких знаний о законе управления формируются непрерывные управляющие воздействия. Это направление хорошо разработано и широко применяется на практике уже более двух десятилетий. Алгоритм состоит в фазификации непрерывных состояний процессов, использовании нечетких правил вывода и последующим применением дефазификации полученных в результате вывода нечетких значений лингвистических переменных [8]. Этот алгоритм может быть использован для реализации транзитивных отношений  $\tau_k(x_k, x)$  в исполнительных частях правил (3). Пример реализации основанного на правилах (1,3) гибридного закона управления приведен в [9]. Второе направление, получившее пока меньшее распространение, состоит в выработке четких решений на основании нечетких знаний [10]. В данной статье рассматривается методология использования нечетких предикатов для принятия решений при моделировании гибридных процессов с помощью трансформационных правил (1,3).

## **2. Формализация нечетких знаний о состояниях процессов.**

Основным источником неопределенности при спецификации гибридных процессов являются нечеткие утверждения о значениях состояний непрерывных процессов (высокое давление, низкая температура, малое рассогласование и т. п.). Такие утверждения являются нечеткими предикатами от соответствующих непрерывно изменяющихся состояний. Степень их истинности TV (Truth Value) определяется значениями из

интервала  $[0, 1]$ . Состояния дискретно-событийных процессов (включен - выключен, открыт - закрыт) и четкие действия или решения представляются значениями четкой двузначной логики (True, False). Условные части правил (3) определяют четкие действия по изменению состояний непрерывных процессов, поэтому в них могут входить только четкие логические переменные. В условные части правил (1) могут входить как четкие, так и нечеткие логические переменные. Если конъюнкция (2), определяющая текущую ситуацию, наряду с четкими переменными содержит хотя бы один нечеткий предикат, то она является нечеткой логико-динамической ситуацией. Для удобства обработки этих правил процедурой интерпретатора желательно иметь единообразное представление степеней истинности как четких, так и нечетких логических переменных. Поскольку нечеткая логика является обобщением четкой классической логики, то принятый в ней способ представления степени истинности с помощью функций принадлежности применим также и для четких логических переменных. Очевидно, что функции принадлежности четких переменных имеют ступенчатый вид. Кроме того, также необходимо для вычисления степеней истинности конъюнкций, дизъюнкций и импликаций использовать формулы пригодные как для четких, так и для нечетких логических переменных. Нетрудно видеть, что таковыми являются min-конъюнкция, max-дизъюнкция и импликация Мамдани [7]:

$$\begin{aligned}
 TV(\tilde{A} \wedge \tilde{B}) &= \min(TV(\tilde{A}), TV(\tilde{B})), \\
 TV(\tilde{A} \vee \tilde{B}) &= \max(TV(\tilde{A}), TV(\tilde{B})), \\
 TV(\tilde{A} \supset \tilde{B}) &= \min(TV(\tilde{A}), TV(\tilde{B})).
 \end{aligned}$$

Эти формулы были использованы для реализации процедуры, обрабатывающей правила (1), при разработке исследовательского прототипа интерпретатора базы правил, содержащей четкие и нечеткие трансформационные правила.

Вторым вопросом, возникающим при реализации указанного интерпретатора правил, является конкретный вид используемых функций принадлежности. Наибольшее применение находят треугольные и трапецидальные функции. Треугольная функция принадлежности может быть задана тремя значениями вещественных чисел  $\{l, c, r\}$ , где  $c$  — значение абсциссы средней вершины, а  $l$  и  $r$  — значения величин отрезков основания треугольника от  $c$  до левой и правой вершины основания, соответственно (рисунок 1).

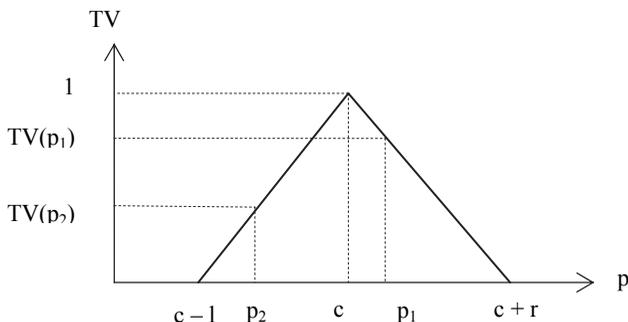


Рис. 1. Задание треугольной функции принадлежности

Вычисление значения степени истинности параметра  $p$  с использованием треугольной функции принадлежности производится следующим образом. Если  $(c-l) < p_2 < c$ , то  $TV(p_2) = \frac{p_2 - c + l}{l}$ . Если  $(c+r) > p_1 > c$ , то  $TV(p_1) = \frac{c + r - p_1}{r}$ . С помощью значений  $\{1, c, r\}$

можно также задавать ступенчатые функции принадлежности в случае четких логических переменных. Для этого надо задать  $c$ , а  $l$  (или  $r$ ) присвоить значение 0 и  $r$  (или  $l$ ) присвоить значение бесконечности. Вместо бесконечности вполне можно использовать одно из имеющихся в компьютере больших чисел (например,  $\text{MaxSingle} = 3,4 \cdot 10^{38}$ ). При этом методическая погрешность вычисления  $TV$  будет пренебрежимо мала. Предикат с трапециoidalной функцией принадлежности можно задать с помощью конъюнкции двух предикатов, имеющих функции принадлежности, показанные на рисунке 2.

Трансформационное правило (1) представляет собой продукционное правило логического вывода Modus Ponens, которое на основании истинности конъюнкции посылок (левая часть правила) и истинности импликации позволяет сделать вывод об истинности заключения (правая часть правила), то есть  $p \wedge (p \rightarrow q) \Rightarrow q$ . Таким образом, в случае нечеткого вывода для того, чтобы получить оценку истинности заключения, необходимо знать степень истинности самого правила. В системах нечеткого вывода каждому правилу присваивается коэффициент определенности или уверенности нечеткой продукции  $F$ . Этот коэффициент определяется экспертным путем из интервала  $[0, 1]$  и используется также как оценки истинности условия правила.

Так, если имеем правило  $s_1, s_2 \rightarrow q, F$ , то  $TV(q) = \min(s_1, s_2, F)$ . Если в базе правил имеется несколько правил, определяющих одно и то же заключение, то эти правила образуют друг с другом дизъюнкцию и, следовательно, степени истинности данного заключения необходимо присвоить максимальное из значений, полученных при обработке каждого из этих правил. Описанный алгоритм обработки правил (1) используется в случае, когда с помощью этих правила необходимо определять степень истинности нечетких ситуаций.

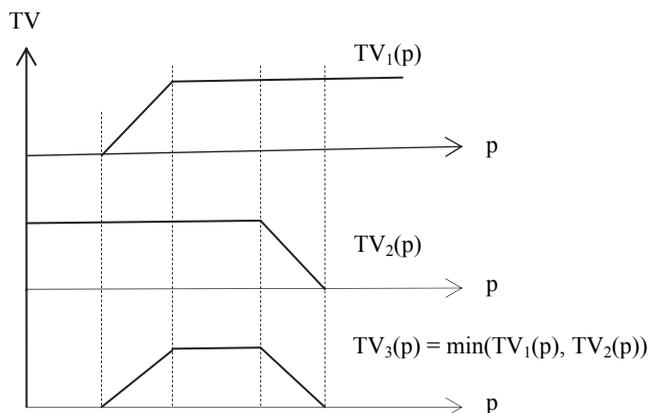


Рис. 2. Формирование трапециевидальной функции принадлежности

Кроме этого, эти правила также используются для формирования четких действий (решений) на основании оценки степени истинности условия. В этом случае правилу также присваивается некоторое значение степени истинности  $F$ , но его смысл и алгоритм использования отличаются от рассмотренного случая оценки истинности ситуаций. В данном случае  $F$  означает пороговое значение истинности условия, при котором считается необходимым принять решение, указанное в исполнительной части правила. Другими словами, если степень истинности условия больше или равна  $F$ , то степени истинности заключения присваивается значение 1.

**2. Иллюстративные примеры использования нечетких предикатов для реализации гибридных процессов.** С целью экспериментальной проверки методов спецификации нечетких предикатов и использования правил нечеткого логического вывода был разработан исследовательский прототип интерпретатора нечетких трансформационных правил. Это было сделано путем модификации разработанного ранее интерпретатора четких правил [11]. При этом из соображений

большей надежности реализации процедур для оценок степени истинности TV логических переменных использовался не принятый в теории интервал вещественных чисел  $[0, 1]$ , а интервал целых положительных чисел  $[0, 1000]$ . Помимо базы правил трансформации нечетких ситуаций была введена база нечетких правил трансформации четких действий и разработаны процедуры обработки соответствующих правил. При обработке правил трансформации ситуаций и действий реализовано следующее условие. Если в ходе данного цикла обработки правил нет ни одного правила, присваивающего некоторой переменной определенное значение степени истинности, то ей присваивается наименьшее значение, то есть 0. Это означает, что отсутствие конкретного значения переменной трактуется как полное ее отрицание. Применение такого условия “по умолчанию” позволяет упростить разработку непротиворечивых баз правил. Рассматриваемые далее модели процессов были реализованы с помощью этого интерпретатора. Используемые трансформационные правила и процедуры спецификации нечетких предикатов представлены в форматах редакторов интерпретатора.

**2.1. Процесс управления уровнем жидкости.** Система состоит из емкости, впускного и выпускного клапанов. В исходном состоянии имеется некоторый уровень жидкости. Выпускной клапан всегда открыт, поэтому уровень понижается со скоростью пропорциональной уровню. Когда уровень достигает некоторого низкого значения, открывается впускной клапан. Скорость поступления жидкости постоянна и превышает скорость вытекания. Поэтому уровень повышается. Когда уровень достигает некоторого высокого значения, впускной клапан закрывается, и процесс изменения уровня повторяется. Для спецификации этой модели процессов надо ввести непрерывную переменную для представления уровня (*Уровень  $L$* ), задать скорость повышения уровня за счет открытого впускного клапана (*Скорость  $V$* ) и коэффициент пропорциональности между величиной уровня и скоростью его понижения за счет открытого выпускного клапана. Необходимо ввести логическую переменную для представления состояния впускного клапана (*Вх. клапан открыт*) и нечеткие предикаты (*Низкий уровень*) и (*Высокий уровень*). На рисунке 3 представлены правила (типа (3)) для реализации изменений уровня. В редакторе правил имя переменной представляется на белом фоне, если используется сама переменная, и — на черном фоне, если используется отрицание переменной.

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация
1	Уровень L	Интеграл	1.00	Скорость V		Вх. клапан открыт
2	Уровень L	Интеграл	-0.10	Уровень L		Ever True

Рис. 3. Правила спецификации процесса изменения уровня

Первое правило определяет повышение уровня со скоростью *Скорость V* в ситуации, когда открыт впускной клапан. Второе правило обуславливает понижение уровня со скоростью пропорциональной уровню. Оно срабатывает всегда, так как EverTrue обозначает логическую константу, имеющую значение True (в нашем интерпретаторе  $TV(EverTrue) = 1000$ ). Очевидно, когда впускной клапан открыт состояние уровня определяется обоими правилами. Нечеткие предикаты заданы функциями принадлежности, представленными на рисунке 4.

№	Предикат	Аргумент 1	< / >	Левое значение (-)	Аргумент 2	+	Константа	Правое значение (+)
1	Низкий уровень	Уровень L	<	3.400000000E+0038		+	100.00	30.00
2	Высокий уровень	Уровень L	>	30.00		+	300.00	3.400000000E+0038

Рис. 4. Спецификация нечетких предикатов системы управления уровнем

В соответствие с этими функциями  $TV(\text{Низкий уровень}) = 1000$  при значениях уровня от 0 до 100 и  $TV(\text{Низкий уровень}) = 0$  при уровне больше 130. В свою очередь  $TV(\text{Высокий уровень}) = 0$  при изменении уровня от 0 до 270 и  $TV(\text{Высокий уровень}) = 1000$  при значениях *Уровень L* = 300 и выше.

Задача управления состоит в том, чтобы открывать впускной клапан, когда уровень низкий и закрывать его, когда уровень высокий. Текущая ситуация в системе определяется значениями двух нечетких предикатов и состоянием впускного клапана, которое представляется значениями четкой логической переменной *Вх. клапан открыт*. На рисунке 5 приведены правила управления впускным клапаном. Условные части правил содержат по одной четкой и одной нечеткой переменной. Черный фон имени переменной на рисунке обозначает отрицание этой переменной.

1	если	$TV \geq$	1000	Низкий уровень	<b>Вх. клапан открыт</b>
	то	$TV =$	1000	Вх. клапан открыт	
2	если	$TV \geq$	1	Вх. клапан открыт	<b>Высокий уровень</b>
	то	$TV =$	1000	Вх. клапан открыт	

Рис. 5. Правила спецификации действий системы управления уровнем

Первое правило специфицирует ситуацию, при которой клапан должен быть открыт. При указанном значении  $TV$  этого правила клапан откроется, когда  $TV(\text{Низкий уровень}) = 1000$ , то есть в соответствие со своей функцией принадлежности, когда  $\text{Уровень } L = 100$ . При открытии клапана уровень начинает повышаться, и первое правило перестает срабатывать. Одновременно с этим начинает срабатывать второе. Это правило определяет открытие клапана до тех пор, пока степень истинности отрицания предиката *Высокий уровень* не достигнет указанного в правиле значения 1. А это произойдет тогда, когда степень истинности самого предиката будет равна 999, то есть, когда  $\text{Уровень } L \approx 300$ . При этом ни одно из правил не будет срабатывать и в соответствие с принятым правилом (по умолчанию) степени истинности четкой переменной *Вх. клапан открыт* будет присвоено значение 0, что соответствует закрытию клапана. На рисунке 6 приведен график процесса, специфицированного рассмотренными правилами.

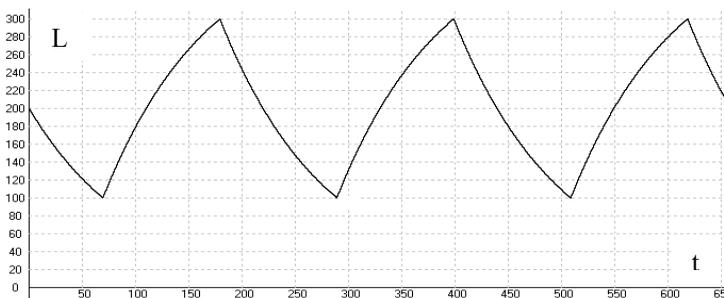


Рис. 6. График процесса управления уровнем

Процесс представлен отрезками экспонент, так как в обоих случаях скорость изменения уровня пропорциональна уровню. В данном простом случае по характеру функций принадлежности предикатов нетрудно заключить, что при изменении  $TV$  правил (рис 5) от 1 до 1000, уровень, при котором открывается клапан, будет изменяться от 100 до 130, а уровень, при котором он закрывается, будет изменяться от 270 до 300.

**2.2. Процесс позиционирования шарика на плоскости.** Шарик находится на плоскости. Система управления может изменять наклон плоскости вокруг одной оси. Задача состоит в том, чтобы путем управления наклоном плоскости перевести шарик из исходного положения в целевое за минимальное время. Ограничением является максимально допустимый угол наклона плоскости. Дополнительным требованием является отсутствие перерегулирования и колебаний в ус-

тойчивом состоянии. Предполагается, что имеются данные об отклонении и о скорости шарика. Оптимальным по быстродействию в данном случае является релейный закон управления, при котором вначале плоскость наклоняется на максимальный угол в сторону цели, а затем при достижении шариком определенного отклонения наклон плоскости изменяется на максимальный противоположный. Значение этого отклонения  $E_{реле}$  зависит от скорости и угла наклона, а именно,

$$E_{реле} = V^2 / g \cdot \sin(\alpha), \text{ где } g \text{ — ускорение свободного падения, } \alpha \text{ —}$$

угол наклона плоскости. Однако при реализации этого закона в окрестности цели возникают колебания угла наклона плоскости, что недопустимо. Поэтому в некоторой небольшой окрестности цели необходимо переключить релейный закон управления на один из пропорциональных законов, например, на ПИД. Это переключение можно задать с помощью нечеткого предиката *Окрестность цели*. Изменение угла наклона в релейном законе управления реализуется с помощью четких предикатов. На рисунке 7 представлена спецификация предикатов рассматриваемой модели системы.

№	Предикат	Аргумент 1	< / >	Левое значение (-)	Аргумент 2	+	Константа	Правое значение (+)
1	$V > 0$	Скорость	>	0.00		+	0.00	3.400000000E+0038
2	$V < 0$	Скорость	<	3.400000000E+0038		+	0.00	0.00
3	$E > E_{релеP}$	$E = X_{цель} - X$	>	0.00	$E_{реле P}$	+	0.00	3.400000000E+0038
4	$E < E_{релеN}$	$E = X_{цель} - X$	<	3.400000000E+0038	$E_{реле N}$	+	0.00	0.00
5	Отклонение 1	$E = X_{цель} - X$	>	5.00		+	-10.00	3.400000000E+0038
6	Отклонение 2	$E = X_{цель} - X$	<	3.400000000E+0038		+	10.00	5.00

Рис. 7. Задание функций принадлежности предикатов

Первые четыре строки определяют функции принадлежности четких предикатов. Эти предикаты обеспечивают реализацию релейного закона управления в случаях положительного и отрицательного начальных отклонений. Последние две строки определяют функции принадлежности нечетких предикатов для положительной и отрицательной частей окрестности. Ситуация, определяющая окрестность цели, задается конъюнкцией этих нечетких предикатов в соответствии с правилом (1) (рисунк 8).

1 если	Rule TV =	1000	Отклонение 1	Отклонение 2	
то	TV =	min	Окрестность цели		

Рис. 8. Правило формирования ситуации *Окрестность цели*

Рассмотренные предикаты используются в правилах трансформации действий, приведенных на рисунке 9. Условные части первых четырех правил содержат четкие предикаты. Специфицированные этими правилами действия предназначены для реализации релейных изменений угла наклона плоскости. Пятое правило специфицирует действие *ПИД управление*, которое необходимо выполнить в ситуации, когда шарик находится в окрестности цели.

1	если	$TV \geq 800$	$E > \text{ЕрелеP}$	$V > 0$
	то	$TV = 1000$	$U\_NB$	
2	если	$TV \geq 800$	$E > \text{ЕрелеF}$	$V > 0$
	то	$TV = 1000$	$U\_PB$	
3	если	$TV \geq 800$	$E < \text{ЕрелеN}$	$V < 0$
	то	$TV = 1000$	$U\_PB$	
4	если	$TV \geq 800$	$E < \text{ЕрелеN}$	$V < 0$
	то	$TV = 1000$	$U\_NB$	
5	если	$TV \geq 1000$	Окрестность цели	
	то	$TV = 1000$	ПИД управление	

Рис. 9. Правила трансформации действий по управлению углом

И, наконец, на рисунке 10 представлены правила (3), специфицирующие непрерывные изменения состояний модели. Первые четыре правила определяют изменения ускорения, скорости и координаты шарика в соответствие с углом наклона плоскости. Пятое и шестое правила вычисляют значение отклонения, при котором необходимо произвести релейное изменение угла. Седьмое правило производит вычисление текущего отклонения, необходимого для спецификации предикатов (рисунок 7). Восьмое и девятое правила реализуют релейный закон управления, а последнее правило — ПИД закон управления. При запуске рассмотренных правил на исполнение интерпретатором были реализованы процессы изменений угла, скорости и координаты. Графики этих процессов представлены на рисунке 11.

№	Переменная	Процедура	Кoeffициент	Аргумент 1	Аргумент 2	Ситуация
1	$\text{Sin}(\text{yгол})$	Синус	1.00	Угол		Ever True
2	Ускорение	Пропорциональная	-490.50	$\text{Sin}(\text{yгол})$		Ever True
3	Скорость	Интеграл	1.00	Ускорение		Ever True
4	X координата	Интеграл	1.00	Скорость		Ever True
5	Е реле P	Умножение	0.0020387	Скорость	Скорость	Ever True
6	Е реле N	Пропорциональная	-1.00	Е реле P		Ever True
7	$E = X \text{ цель} \cdot X$	Сумма / разность	1.00	X цель	X координата	Ever True
8	Угол	Пропорциональная	1.00	Угол_PB		U_PB
9	Угол	Пропорциональная	1.00	Угол_NB		U_NB
10	Угол	ПИД	-1.00	$E = X \text{ цель} \cdot X$		ПИД управление

Рис. 10. Правила трансформации непрерывных состояний процессов

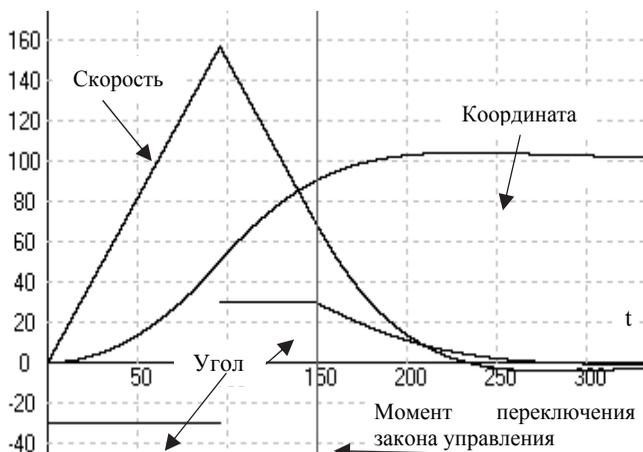


Рис. 11. Графики процессов в системе позиционирования шарика

Из графиков видно, что значение угла наклона плоскости изменяется скачком в соответствие с релейным законом управления до момента переключения на ПИД закон управления, после чего плавно изменяется до нуля. Скорость изменяется в соответствие с изменением угла наклона, а координата — в соответствие со скоростью.

**5. Заключение.** Рассмотрена методология формализации нечетких логико-динамических ситуаций, содержащих как четкие логические переменные, так и нечеткие предикаты от непрерывных состояний процессов. Рассмотрение ограничено случаем использования для спецификации нечетких ситуаций *min*-конъюнкции, *max*-дизъюнкции и импликации Мамдани, а также использованием для спецификации нечетких предикатов треугольных функций принадлежности. Проведенное экспериментальное исследование методики с помощью разработанного интерпретатора нечетких трансформационных правил подтвердило возможность ее практического применения. Рассмотрение примеров моделирования преследовало только цель иллюстрации методики. Практического значения эти модели не имеют. Дальнейшие исследования целесообразно проводить в направлении моделирования биофизических и биохимических систем, в описаниях которых часто присутствует значительное количество нечетких знаний [12].

### Литература

1. *Witsch M., Vogel-Heuser B.* Towards a formal specification framework for manufacturing execution systems // IEEE Transactions on Industrial Informatics. 2012. vol. 8. no. 2. pp. 311–320.
2. *Casagrande A., Piazza C., Policriti A.* Discrete Semantics for Hybrid Automata // Discrete Event Dynamic Systems. 2009. vol. 19. no. 4. pp. 471–493.

3. *Zhirabok A.N., Suvorov A.Yu., Shumskii A.E.* Robust diagnosis of discrete systems with delay: Logic-dynamical approach // *Journal of Computer and Systems Sciences International*. 2014. vol. 53. no. 1. pp. 47–62.
4. *Шпаков В.М.* Исполняемые спецификации транзитивных моделей технологических процессов // *Мехатроника, автоматизация, управление*. 2004. № 3. С. 38–45.
5. *Буткевич И.К., Рыдник Е. А., Шпаков В. М.* Использование среды EnviCon для моделирования производства жидкого гелия // *Труды СПИИРАН*. 2007. Вып. 5. С. 328–332.
6. *Шпаков В.М.* Об использовании трансформационных правил для компьютерной реализации непрерывных процессов // *Труды СПИИРАН*. 2014. Вып. 33. С. 99–116.
7. *Ross T.J.* *Fuzzy Logic with Engineering Applications (Third Edition)* // John Wiley & Sons. 2010. 585 p. URL: <http://onlinelibrary.wiley.com/book/10.1002/9781119994374>.
8. *Григорьев М.В., Кульков С.Н.* Нечеткий логический вывод в системе управления беспилотного летательного аппарата // *Журнал Сибирского федерального университета*. 2011. Т. 4. № 1. С. 79–91.
9. *Шпаков В.М.* Пример спецификации гибридного логического закона управления // *Материалы 2-й Российской мультиконференции по проблемам управления*. Мехатроника, автоматизация, управление. С.-Петербург: ГНЦ РФ ЦНИИ «Электронприбор». 2008. С. 133–136.
10. *Verma O.P., Jain V., Gumber R.* Simple fuzzy rule based edge detection // *Journal of Information Processing Systems*. 2013. vol. 9. no. 4. pp. 575–591.
11. *Шпаков В.М.* Прототип среды моделирования структурированных совокупностей взаимодействующих процессов // *Сборник докладов конференции “Имитационное моделирование. Теория и практика”*. Санкт-Петербург. 2005. Т. II. С. 292–295.
12. *Wang Y., Hu X.* Fuzzy reasoning of accident provenance in pervasive healthcare monitoring systems // *IEEE Journal of Biomedical and Health Informatics*. 2013. vol. 17. no. 6. pp. 1015–1022.

## References

1. Witsch M., Vogel-Heuser B. Towards a formal specification framework for manufacturing execution systems. *IEEE Transactions on Industrial Informatics*. 2012. vol. 8. no. 2. pp. 311–320.
2. Casagrande A., Piazza C., Policriti A. Discrete Semantics for Hybrid Automata. *Discrete Event Dynamic Systems*. 2009. vol. 19. no. 4. pp. 471–493.
3. Zhirabok A.N., Suvorov A.Yu., Shumskii A.E. Robust diagnosis of discrete systems with delay: Logic-dynamical approach. *Journal of Computer and Systems Sciences International*. 2014. vol. 53. no. 1. pp. 47–62.
4. Shpakov V.M. [Executable Specifications of production process transitive models]. *Mekhatronika, avtomatizatsiya, upravlenie – Mechatronics, automation, control*. 2004. № 3. С. 38–45. (In Russ.).
5. Butkevich I.K., Rydник E.A., Shpakov V.M. [Using environment EnviCon for liquid helium production simulation]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2007. vol. 5. pp. 328–332.
6. Shpakov V.M. [About use of transformation rules for computer implementation of continuous processes]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2014. vol. 33. pp. 99–116. (In Russ.).
7. Ross T.J. *Fuzzy Logic with Engineering Applications (Third Edition)*. John Wiley & Sons. 2010. 585 p. URL: <http://onlinelibrary.wiley.com/book/10.1002/9781119994374>.

8. *Grigorev M.V., Kulkov S.N.* [Fuzzy logical deduction in control system of an unmanned aerial vehicle]. *Zhurnal Sibirskogo federalnogo universiteta – Journal of Siberian Federal University*. 2011. vol. 4. no. 1. pp. 79–91. (In Russ.).
9. *Shpakov V.M.* [An example of hybrid logic control law specification]. *Materialy 2-oy multiconferentsii po problemam upravleniya. Mekhatronika, avtomatizatsiya, upravlenie* [The materials of the 2nd Russian multiconference on control issues. Mechatronics, automation, control]. S.-Peterburg: GNTs RF TsNII “Elektropribor”. 2008. pp. 133–136. (In Russ.).
10. *Verma O.P., Jain V., Gumber R.* Simple fuzzy rule based edge detection. *Journal of Information Processing Systems*. 2013. vol. 9. no. 4. pp. 575–591.
11. *Shpakov V.M.* [Prototype of an environment for interacting processes structured collection simulation]. *Sbornik dokladov konferentsii "Imitatsionnoe modelirovanie. Teorija i praktika"* [Conference Proceedings "Simulation. Theory and practice"] St. Petersburg. 2005. vol. II. pp. 292–295. (In Russ.).
12. *Wang Y., Hu X.* Fuzzy reasoning of accident provenance in pervasive healthcare monitoring systems. *IEEE Journal of Biomedical and Health Informatics*. 2013. vol. 17. no. 6. pp. 1015–1022.

**Шпаков Владимир Михайлович** — к-т техн. наук, доцент, старший научный сотрудник, лаборатория интегрированных систем автоматизации Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: автоматическое управление, искусственный интеллект, логическое программирование, экспертные системы, поддержка принятия решений. Число научных публикаций — 70. vlad@iias.spb.su; 14-я линия В.О., д. 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-8071, факс +7(812)328-4450

**Shpakov Vladimir Michajlovich** — Ph.D., associate professor, senior researcher, laboratory of integrated systems for automation of St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: automatic control, artificial intelligence, logic programming, expert systems, decision making support. The number of publications — 70. vlad@iias.spb.su; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-8071, fax +7(812)328-4450

**Поддержка исследований.** Работа выполнена при финансовой поддержке РФФИ (проект № 12-01-00015-а).

**Acknowledgements.** This research is supported by RFBR (grant #12-01-00015-a).

## РЕФЕРАТ

### **Шпаков В.М. Формализация и использование нечетких предикатов для реализации гибридных процессов.**

Функционирование многих динамических систем может быть представлено совокупностью взаимодействующих гибридных процессов. Потребность в компьютерной реализации таких совокупностей процессов или их частей возникает при моделировании указанных систем и при компьютерной реализации систем управления ими. В статье рассматривается методика реализации гибридных процессов, основанная на использовании трансформационных правил для спецификации процессов. Методика отличается от описанных ранее тем, что наряду с четкими логическими переменными позволяет использовать нечеткие предикаты для спецификации нечетких логико-динамических ситуаций и четких логических команд (действий, решений).

Методика предполагает спецификацию четких и нечетких логических переменных с помощью треугольных и трапецеидальных функций принадлежности. Показано, что ступенчатые функции принадлежности, необходимые для спецификации четких логических переменных, могут быть представлены с помощью вырожденных треугольных функций, то есть таких, у которых одна из вершин основания треугольника находится в бесконечности. На практике координате такой вершины присваивается очень большое числовое значение. Логические переменные с трапецеидальной функцией принадлежности формируются с помощью конъюнкции переменных, имеющих треугольные функции. Для выполнения необходимых при обработке правил вычислений логических функций предлагается использовать алгоритмы, пригодные для четких и нечетких переменных. Помимо правил трансформации нечетких ситуаций методика предусматривает использование правил, формирующих четкие действия. В этих правилах устанавливается порог истинности. Если степень истинности ситуации в условной части правила превышает заданный порог, то выполняется действие, специфицированное правой частью правила.

Экспериментальное исследование методики было проведено с помощью разработанного интерпретатора трансформационных правил, допускающего использование как четких, так и нечетких логических переменных. В качестве примера приведена спецификация с помощью трансформационных правил системы управления уровнем. Спецификация содержит два нечетких предиката: *Высокий уровень* и *Низкий уровень*. В качестве второго примера рассмотрена спецификация системы позиционирования шарика на плоскости. В спецификации использован нечеткий предикат *Окрестность цели*, имеющий трапецеидальную функцию принадлежности. Этот предикат определяет переключение закона управления углом наклона плоскости при достижении малого отклонения от цели. Проведенное моделирование систем и полученные процессы позволяют сделать вывод о работоспособности рассмотренной методики.

## SUMMARY

### *Shpakov V.M.* **Formalization and Use of Fuzzy Predicates for Hybrid Processes Implementation.**

The functioning of many dynamic systems can be represented by a set of interacting hybrid processes. The need for computer implementation of such sets of processes or parts thereof occurs when modeling these systems and computer implementation of their control systems. In the article the method of implementation of hybrid processes based on the use of transformation rules for the processes specification are considered. The method differs from previously described by the fact that along with a crisp logical variables, allows the use of fuzzy predicates for the specification of fuzzy logic-dynamic situations and crisp logical commands (actions, decisions).

The technique involves the specification of crisp and fuzzy logical variables using triangular and trapezoidal membership functions. It is shown that the step function facilities required for the specification of crisp logical variables can be represented using degenerate triangular functions, i.e. those in which one of the vertices of the triangle's base is at infinity. In practice, the coordinate of this vertex is assigned a very large numerical value. Logical variables with trapezoidal membership functions are formed using the conjunction of the variables that have triangular functions. To perform when processing rules for computing logical functions is proposed to use algorithms that are suitable for crisp and fuzzy variables. In addition to the transformation rules of fuzzy situations, the technique involves the use of rules, forming crisp actions. A truth threshold of the condition is set in these rules. If the degree of truth of the situation in the conditional part of the rule exceeds the specified threshold, then the action, specified by the right-hand side of the rule is executed.

Experimental research of the method was carried out using the developed interpreter of transformation rules that allow the use of both crisp and fuzzy logic variables. As an example, a specification of level control system by means of transformation rules is presented. The specification contains two fuzzy predicates: *High level* and *Low level*. As a second example, the specification of the system positioning the ball on the plane is considered. In the specification it was used fuzzy predicate *Goal vicinity* having a trapezoidal membership function. This predicate defines the switching of a control law of the plane angle inclination when the ball achieves a small deviation from the target. The systems simulation and the resulting processes allow us to conclude about the efficiency of the techniques considered.