

Лекция 9

Язык нечеткого управления – FCL

Язык нечеткого управления FCL (Fuzzy Control Language) описан в Стандарте ИЕС 1131-7, в котором определяются цели разработки этого языка, его базовая нотация и приводятся примеры записи моделей нечеткого управления с использованием нотации языка FCL. В настоящей главе рассматриваются элементы языка FCL в соответствии с указанным Стандартом.

Язык FCL разработан для представления нечетких моделей систем управления, в частности, моделей так называемых программируемых контроллеров (Programmable Controllers) или *программируемых логических контроллеров* (ПЛК) в форме структурируемого текста, который может быть интерпретирован как программа на языке высокого уровня. Хотя Стандарт ИЕС 1131-7 не определяет требования к вычислительным средам и устройствам, которые могут реализовывать трансляцию, компиляцию и выполнение программ на языке FCL, описанная в нем нотация основных компонентов систем нечеткого вывода позволяет достичь формального уровня строгости, необходимого для последующей разработки соответствующих инструментальных средств.

В лекции рассматриваются элементы базовой нотации языка FCL. В заключение этой главы подробно рассматривается конкретный пример разработки нечеткой модели системы управления смесителем воды при принятии душа и запись этой модели в нотации языка FCL.

9.1. Базовая нотация языка нечеткого управления FCL

В настоящем разделе рассматриваются элементы языка FCL, их синтаксис и правила использования в нечетких моделях систем управления. Описание этих элементов занимает центральное место в Стандарте ИЕС 1131-7.

Основные элементы языка FCL

Метод спецификации текстового описания программируемых контроллеров приводится в Приложении А Стандарта ИЕС 1131-3 (ИЕС 1131 Part 3: 1993, Programming languages). Этот же метод спецификации используется при описании языка FCL.

Формальная спецификация элементов языка для текстовых языков программирования описывается в Приложении В Стандарта ИЕС 1131-3. Для языка FCL следует использовать следующее подмножество этих элементов языка:

- Буквы, цифры, идентификаторы
- Константы
- Типы данных
- Переменные

Нотация правил продукций

В дополнение к перечисленным выше элементам языка Части 3 Стандарта ИЕС 1131 для записи правил продукций следует использовать следующие элементы языка FCL:

```
объявление_функционального_блока ::= 'FUNCTION_BLOCK'  
имя_функционального_блока  
  {объявления_вх_вых_переменных_ФБ}  
  {объявления_других_переменных_ФБ}  
тело_функционального_блока  
'END_FUNCTION_BLOCK'  
объявления_вх_вых_переменных_ФБ ::= объявления_входа | объявления_выхода  
объявления_других_переменных_ФБ ::= объявления_переменных  
тело_функционального_блока ::= {блок_фаззификации}
```

```

    {блок_дефаззификации}
    {блок_правил}
    {необязательный_блок}
блок_фааззификации ::= 'FUZZIFY' имя_переменной
    {лингвистический_терм}
'END_FUZZIFY'
блок_дефаззификации ::= 'DEFUZZIFY' имя_нечеткой_переменной
    {лингвистический_терм}
    метод_дефаззификации
    значение_по_умолчанию
    [интервал]
    'END_FUZZIFY'
блок_правил ::= 'RULEBLOCK' имя_блока_правил
    определение_операции
    [метод_активизации]
    метод_аккумуляции
    {правило}
'END_RULEBLOCK'
необязательный_блок ::= 'OPTION'
любой_параметр_спецификации_модели
'END_OPTION'
лингвистический_терм ::= 'TERM' имя_терма ':'= функция_принадлежности ';'
функция_принадлежности ::= одноэлементное_множество | точки
одноэлементное_множество ::= число | имя_переменной
точки ::= {'('число | имя_переменной',' число')'}
метод_дефаззификации ::= 'METHOD* ':' 'COG' | 'COGS' | 'COA' | 'LM' | 'RM';
значение_по_умолчанию ::= 'DEFAULT' ':'= число | 'NC' ';';
интервал ::= 'RANGE' ':'= '('число'..'число')' ';';
определение_операции ::= ('OR' ':' 'MAX' | 'ASUM' | 'BSUM') | ('AND' ':'
'MIN' | 'PROD' | 'BDIF') ';';
метод_активизации ::= 'ACT' ':' 'PROD' | 'MIN' ';';
метод_аккумуляции ::= 'ACCU' ':' 'MAX' | 'BSUM' | 'NSUM' ';';
правило ::= 'RULE' целое_число ':'
'IF' условие 'THEN' заключение [WITH весовой_коэффициент] ';';
условие ::= (подусловие | имя_переменной) {'AND' | 'OR' подусловие |
имя_переменной}
подусловие ::= ('NOT' '('имя_переменной' 'IS' ['NOT']) имя_терма') |
(имя_переменной 'IS' ['NOT'] имя_терма)
заключение ::= { (имя_переменной | (имя_переменной 'IS' имя_терма)) ', ' }
(имя_переменной | имя_переменной 'IS' имя_терма)
весовой_коэффициент ::= переменная | число
имя_функционального_блока ::= идентификатор
имя_блока_правил ::= идентификатор
имя_терма ::= идентификатор
имя_нечеткой_переменной ::= идентификатор
имя_переменной ::= идентификатор
число ::= целое_число | действительное_число
объявления_входа ::= see IEC 1131-3 Annex B
объявления_выхода ::= see IEC 1131-3 Annex B
объявления_переменных ::= see IEC 1131-3 Annex B
идентификатор ::= see IEC 1131-3 Annex B

```

Ключевые слова языка FCL

В табл. 9.1 приводится перечень ключевых слов языка FCL, которые разрешается использовать только в том значении, как оно определено в нотации языка FCL.

Таблица 9.1. Зарезервированные ключевые слова

Ключевое слово	Описание ключевого слова
()	Скобки в условии, терме, интервале
ACCU	Метод аккумуляции
ACT	Метод активизации

AND	Операция логической конъюнкции (логическое "И")
ASUM	Операция логического "ИЛИ" по формуле алгебраической суммы
Ключевое слово	Описание ключевого слова
BDIF	Операция логического "И" по формуле ограниченной разности
BSUM	Метод аккумуляции по формуле ограниченной суммы
COA	Метод дефаззификации по формуле центра площади
COG	Метод дефаззификации по формуле центра тяжести
COGS	Метод дефаззификации по формуле центра тяжести для одноэлементных множеств
DEFAULT	Значение выхода по умолчанию в случае отсутствия активных правил
FUZZIFY	Фаззификация входной переменной
IF	Начало правила, после которого следует его условие
IS	Связка для значения лингвистической переменной в условии или заключении
LM	Метод дефаззификации по формуле левого модального значения
MAX	Метод аккумуляции по формуле максимума
METHOD	Метод дефаззификации
MIN	Минимум в качестве логической операции "И"
NC	Значение выходной переменной остается без изменения (No Change) в случае отсутствия активных правил
NOT	Операция логического отрицания (логическое "НЕ")
NSUM	Метод аккумуляции по формуле нормализованной суммы
OPTIONS	Начало спецификации дополнительных параметров
OR	Операция логической дизъюнкции (логическое "ИЛИ")
PROD	Произведение в качестве логической операции "И"
RANGE	Интервал переменной для шкалирования функции принадлежности
RM	Метод дефаззификации по формуле правого модального значения
RULE	Начало спецификации нечеткого правила
RULEBLOCK	Начало спецификации блока правил
TERM	Определение лингвистического термина (функции принадлежности) для лингвистической переменной
THEN	Разделитель условия и заключения в нечетком правиле
VAR	Определение локальной переменной (переменных)
VAR_INPUT	Определение входной переменной (переменных)
VAR_OUTPUT	Определение выходной переменной (переменных)
WITH	Определение весового коэффициента

Интерфейс функционального блока (Function Block interface)

Согласно Стандарту IEC 1131-7, внешнее представление Нечеткого Функционального Блока требует использования следующих элементов языка FCL:

```

FUNCTION_BLOCK имя_функционального_блока {Функциональный блок}
VAR_INPUT {Объявление входных параметров}
имя-переменной: тип_данных;
...
END_VAR
VAR_OUTPUT {Объявление выходных параметров}
имя-переменной: тип_данных;
...

```

```

END_VAR
...
VAR      {Локальные переменные}
имя-переменной: тип_данных;
END_VAR
END_FUNCTION_BLOCK

```

Эти элементы языка FCL позволяют описать интерфейс функционального блока. Интерфейс функционального блока определяется совместно с параметрами, которые могут использоваться внутри и вне этого функционального блока. Типы данных этих параметров должны быть определены в соответствии с требованиями Стандарта языка FCL.

Фаззификация (Fuzzification)

Как было сказано в *лекции 8*, цель фаззификации заключается в преобразовании числовых значений входных переменных в функции принадлежности, определенные для лингвистических значений соответствующих переменных. Это преобразование в языке FCL описывается между ключевыми словами: FUZZIFY и END_FUZZIFY следующим образом:

```

FUZZIFY имя_переменной
TERM имя_терма := функции_принадлежности;
...
END_FUZZIFY

```

После ключевого слова FUZZIFY должно быть записано имя переменной, которая используется для фаззификации. Это имя переменной предварительно должно быть определено в секции VARINPUT. Соответствующая лингвистическая переменная должна быть описана одним или несколькими лингвистическими термами.

Лингвистические термы определяются после ключевого слова TERM и описываются посредством функций принадлежности, которые фаззифицируют эту переменную. В качестве функции принадлежности может использоваться некоторая кусочно-линейная функция. Эта функция задается таблицей своих точек в форме:

```

функция_принадлежности ::= (точка i), (точка j), . . .

```

Каждая точка в списке представляет собой пару значений, разделенных запятой. Первое из них — количественное значение переменной, а второе — значение функции принадлежности для этого значения переменной. Эти пары значений заключаются в скобки и отделяются друг от друга запятыми:

```

точка i ::= значение входа i | имя_переменной на входе i, значение функции принадлежности i

```

Таким способом могут быть определены все простые элементы, такие как наклонная прямая и треугольная функция принадлежности. Точки должны записываться в возрастающем порядке значений переменной. Функция принадлежности между последовательными точками является линейной. Остальные значения функции принадлежности для произвольного значения входа вычисляются посредством линейной интерполяции значений функции принадлежности для двух соседних точек.

Число точек может изменяться, но их максимальное количество ограничено согласно правилу 6 классов согласованности.

Ниже приводится пример функции принадлежности с 3 точками для лингвистического термина "теплый":

```

TERM теплый := (10°C, 0.0), (50°C, 1.0), (90°, 0.0);

```

Если значение лингвистической переменной меньше, чем первая базовая точка в таблице задания функции принадлежности, то соответствующее значение функции принадлежности принимается равным значению для этой первой точки в таблице. Если значение лингвистической переменной больше, чем последняя базовая точка в таблице задания функции принадлежности, то соответствующее значение функции принадлежности принимается равным значению для этой последней точки в таблице.

Примеры определения линейной Z-образной функции и линейной S-образной функции принадлежности:

```

FUZZIFY температура;
  TERM холодный := (-10°C, 1), (10°C, 0);
  TERM теплый := (10°C, 0.0), (50°C, 1.0);
END_FUZZIFY

```

Чтобы адаптировать модель нечеткого управления на этапе ее реализации, базовые точки функций принадлежности могут быть модифицированы. Это можно сделать посредством определения дополнительных переменных, которые описываются как входные в функциональном блоке. Эти переменные необходимо объявить в секции VAR_INPUT функционального блока.

Ниже представлен пример задания точек функции принадлежности

```

VAR_INPUT
  ТЕМПЕРАТУРА : REAL; (* ЭТОТ ВХОД ДОЛЖЕН БЫТЬ ФАЗЗИФИЦИРОВАН *)
  ДАВЛЕНИЕ : REAL; (*ЭТОТ ВХОД ДОЛЖЕН БЫТЬ ФАЗЗИФИЦИРОВАН *)
  BP_WARML, BP_WARM2 : REAL; (* ЭТИ ВХОДЫ ПРЕДНАЗНАЧЕНЫ ДЛЯ ИНТЕРАКТИВНОЙ
АДАПТАЦИИ *)
END_VAR
FUZZIFY ТЕМПЕРАТУРА
TERM ТЕПЛЫЙ := (BP_WARML, 0.0), (40°C, 1.0), (BP_WARM2, 0.0);
...
END_FUZZIFY

```

Дефаззификация (Defuzzification)

Лингвистическая переменная должна быть преобразована в количественное значение. Это преобразование описывается между ключевыми словами DEFUZZIFY и END_DEFUZZIFY.

Переменная, которая используется для дефаззификации, должна быть записана после ключевого слова DEFUZZIFY. Имя ЭТОЙ переменной должно быть объявлено ранее в секции VAR_OUTPUT.

```

DEFUZZIFY имя_переменной
  TERM имя_терма := функция_принадлежности;
  метод_дефаззификации;
  значение_по_умолчанию;
  [интервал;]
END_DEFUZZIFY

```

Определение лингвистических термов задается в секции фаззификации. С целью упрощения дефаззификации выходов могут быть использованы специальные функции принадлежности в форме значения для одноэлементного множества. Подобные функции принадлежности описываются посредством единственного значения для лингвистического терма.

Ниже приводится пример определения таких термов.

```

DEFUZZIFY клапан
  TERM открыт_на_слив := -100;
  TERM закрыт := 0;
  TERM открыт_на_наполнение := 100; END_DEFUZZIFY

```

Метод дефаззификации должен быть определен посредством элемента языка МЕТОД после ключевого слова METHOD.

```

METHOD : метод_дефаззификации;

```

В качестве метода дефаззификации могут быть использованы следующие методы (табл. 9.2)

Таблица 9.2. Методы дефаззификации

Ключевое слово	Описание	Формула расчета
COG	Центр тяжести (Centre of Gravity)	(8.9)
COGC	Центр тяжести для одноточечных множеств (Centre of Gravity for Singletons)	(8.10)

COA	Центр площади (Centre of Area)	(8.11)
LM	Левое модальное значение (Left Most Maximum)	(8.12)
RM	Правое модальное значение (Right Most Maximum)	(8.13)

Если значение функции принадлежности равно 0 для всех лингвистических термов некоторой выходной переменной, то это означает отсутствие активных правил для этой переменной. В этом случае дефаззификация не позволит получить адекватный результат. Именно по этой причине целесообразно определить для такой переменной некоторое значение по умолчанию. Это значение будет назначено выходной переменной только в случае отсутствия активных правил. Значение по умолчанию задается после ключевого слова DEFAULT.

```
DEFAULT := значение | NC;
```

После ключевого слова DEFAULT может быть указано значение по умолчанию или ключевое слово NC (no change). Последнее применяется для того, чтобы явно указать, что выход остается без изменения в случае, если отсутствуют активные правила.

Интервал определяется заданием после ключевого слова RANGE некоторых минимального и максимального значений, которые разделяются двумя точками.

```
RANGE := (minimum value., maximum value);
```

Интервал используется для спецификации минимального и максимального значений некоторой выходной переменной. Если функции принадлежности термов выходной переменной заданы в форме одноэлементных множеств, то подобный интервал не может быть задан. В остальных случаях RANGE используется для ограничения каждой функции принадлежности интервалом по каждой из выходных переменных. Этот интервал следует явно определять, чтобы избежать непредсказуемых значений выходных переменных.

Если этот интервал не задан, то принимается интервал по умолчанию для типа данных соответствующей переменной, как это определено в части 3 Стандарта ИЕС 1131-3.

Блок правил (Rule block)

Для определения механизма нечеткого вывода используется один или несколько блоков правил. С целью удобства и предоставления возможности декомпозиции базы правил на отдельные модули разрешается использовать несколько блоков правил. При этом каждый блок правил должен иметь уникальное имя.

Правила должны быть определены между ключевыми словами RULEBLOCK и END_RULEBLOCK следующим образом.

```
RULEBLOCK имя_блока_правил
  определение_операции;
  [ метод_активизации; ]
  метод_аккумуляции;
  правила;
END_RULEBLOCK
```

Внутри блока правил используются нечеткие операции

```
определение_операции ::= операция : алгоритм
```

Чтобы соответствовать законам Де Моргана следует использовать парные алгоритмы для операторов AND и OR, например, если для оператора AND используется алгоритм MIN, то для оператора OR следует использовать алгоритм MAX. Соответствующие парные алгоритмы определены в табл. 9.3.

Таблица 9.3. Парные алгоритмы для операций

Оператор OR (логическое "ИЛИ")		Оператор AND (логическое "И")	
Ключевое слово для алгоритма	Алгоритм	Ключевое слово для алгоритма	Алгоритм
MAX	$\max\{\mu_1(x), \mu_2(x)\}$	MIN	$\min\{\mu_1(x), \mu_2(x)\}$
ASUM	$\frac{\mu_1(x) + \mu_2(x) - \mu_1(x) \cdot \mu_2(x)}{2}$	PROD	$\mu_1(x) \cdot \mu_2(x)$
BSUM	$\min\{1, \mu_1(x) + \mu_2(x)\}$	BDIF	$\max\{0, \mu_1(x) + \mu_2(x) - 1\}$

Ниже представлен пример записи блоков правил:

```

RULEBLOCK первый
  AND : MIN;
  ...
END_RULEBLOCK
RULEBLOCK второй
  AND : PROD;
  ...
END_RULEBLOCK

```

Метод активизации определяется после ключевого слова АСТ следующим образом:

АСТ : метод_активизации;

Методы активизации, которые допустимы в языке FCL, определены в табл. 9.4. При этом следует помнить, что метод активизации не применим для одноэлементного множества.

Таблица 9.4. Методы активизации

Название метода	Ключевое слово для метода активизации	Алгоритм
Произведение	PROD	$\mu_1(x) \cdot \mu_2(x)$
Минимум	MIN	$\min\{\mu_1(x), \mu_2(x)\}$

Метод аккумуляции определяется после ключевого слова ACCU следующим образом:

ACCU : метод_аккумуляции;

Методы аккумуляции, которые допустимы в языке FCL, определены в табл. 9.5.

Таблица 9.4. Методы аккумуляции

Название метода	Ключевое слово для метода аккумуляции	Формула
Максимум	MAX	$\max\{\mu_1(x), \mu_2(x)\}$
Ограниченная сумма	BSUM	$\min\{1, \mu_1(x) + \mu_2(x)\}$
Нормализованная сумма	NSUM	$\frac{\mu_1(x) + \mu_2(x)}{\max\{1, \max_{y \in X} \{\mu_1(y) + \mu_2(y)\}\}}$

Входами блока правил являются лингвистические переменные со своими множествами лингвистических термов. Каждый из термов имеет соответствующие значения функции принадлежности.

Правила определяются внутри блока правил. Каждое правило должно иметь в качестве своего имени уникальный номер внутри блока. Этот номер записывается после ключевого слова RULE и заканчивается двоеточием.

```

RULE номер_правила : IF условие THEN заключение
[WITH весовой_коэффициент];

```

Само правило должно начинаться ключевым словом IF, после которого записывается условие правила. После условия должно следовать заключение, которое записывается после ключевого слова THEN.

В одном правиле допускается использовать несколько отдельных подусловий и входных переменных, чтобы иметь возможность оперировать различными нечеткими степенями принадлежности, которые могут быть включены в нечеткий функциональный блок FFB. Все такие подусловия должны быть записаны между ключевыми словами IF и THEN и соединены логическими операциями с ключевыми словами AND, OR, NOT.

Приоритет этих логических операций определяется согласно табл. 9.6.

Таблица 9.6. Приоритет логических операций

Приоритет	Операция	Приоритет	Операция
1	Скобки ()	3	AND (И)
2	NOT (НЕ)	4	OR (ИЛИ)

Ниже приводится простой пример записи некоторого правила:

RULE 1 : IF подусловие1 AND переменная1 OR переменная2 THEN заключение ;

В соответствии с Базовым Уровнем согласованности операция OR в условии некоторого правила может быть заменена определением двух отдельных правил. Например, правило

RULE 3 : IF подусловие1 OR подусловие2 THEN заключение ;

можно заменить на правила:

RULE 3a : IF условие1 THEN заключение ;

RULE 3b : IF условие2 THEN заключение ;

Подусловия начинаются с имени некоторой лингвистической переменной, после которого следует ключевое слово IS с необязательным ключевым словом NOT. После этих ключевых слов записывается один лингвистический терм для лингвистической переменной, которая определяется в соответствующем условии:

подусловие := лингвистическая_переменная IS [NOT] лингвистический_терм

Ниже приводятся примеры записи подусловий в форме структурируемого текста:

температура IS горячая

температура IS NOT горячая

Допускается использование ключевого слова NOT перед записью подусловия. В этом случае следует применять круглые скобки, например:

IF NOT (температура IS горячий) THEN...

Заключение может быть разделено на несколько подзаключений и выходные переменные. Подзаключения начинаются с имени некоторой лингвистической переменной, после которого следует ключевое слово IS с одним лингвистическим термом для этой лингвистической переменной:

подзаключение := лингвистическая_переменная IS лингвистический_терм

Ниже приводится пример нескольких подзаключений, записанных в одну строку:

IF температура IS холодная AND давление IS низкое THEN переменная1, клапан1 IS открыт_на_наполнение, клапан2 IS закрыт ;

или в несколько строк:

IF температура IS холодная AND давление IS низкое

THEN переменная1,

клапан1 IS открыт_на_наполнение,

клапан2 IS закрыт ;

При необходимости, но не обязательно, для каждого подзаключения можно задать весовой коэффициент, который представляет собой некоторое число типа данных REAL и со значением из интервала [0, 1]. Это число записывается после ключевого слова WITH:

IF условие THEN подзаключение1 [WITH весовой_коэффициент], подзаключение2 ;

Выражение предназначено для уменьшения степени принадлежности (значения функции принадлежности) подзаключения посредством умножения результата подзаклучения на этот весовой коэффициент.

Чтобы иметь возможность внешнего изменения параметров приложений нечеткого управления, весовой коэффициент может быть задан как переменная. При этом соответствующая переменная должна быть объявлена в секции VAR_INPUT. Это предоставит возможность изменить весовой коэффициент в ходе выполнения программы и тем самым адаптировать программу нечеткого управления к особенностям решаемой задачи.

Если подзаклучение не имеет записи с ключевым словом WITH, то используется значение весового коэффициента по умолчанию, которое равно 1. Ниже приводится пример записи постоянного весового коэффициента:

```
IF температура IS холодная AND давление IS низкое THEN клапан1 IS открыт_на_наполнение WITH 0.5, клапан2 IS закрыт;
```

Пример записи переменного весового коэффициента:

```
VAR_INPUT
вес_правила1 :REAL := 0.8;
END_VAR
RULEBLOCK правило_для_температуры
  RULE 1: IF температура IS холодная AND давление IS низкое
  THEN клапан1 IS открыт_на_наполнение WITH вес_правила1,
  ...
END RULEBLOCK
```

Простой пример записи модели нечеткого управления с использованием нотации языка FCL

```
FUNCTION_BLOCK Нечеткий_ФБ
VAR_INPUT
  температура : REAL;
  давление : REAL;
END_VAR
VAR_OUTPUT
  клапан : REAL;
END_VAR
FUZZIFY температура
  TERM холодная := (-10°C, 1.0) (10°C, 0.0);
  TERM горячая := (20°C, 0.0) (70°C, 1.0);
END_FUZZIFY
FUZZIFY давление
  TERM низкое := (55, 1.0) (95, 0.0);
  TERM высокое := (55, 0.0) (95, 1.0); END_FUZZIFY
DEFUZZIFY клапан
  TERM открыт_на_слив := -100;
  TERM закрыт := 0;
  TERM открыт_на_наполнение := 100;
  ACCU : MAX;
  METHOD : COGS;
  DEFAULT := 0;
END_DEFUZZIFY
RULE_BLOCK номер_1
  AND : MIN;
  RULE 1 : IF температура IS холодная AND давление IS низкое
  THEN клапан IS открыт_на_слив
  RULE 2 : IF температура IS холодная AND давление IS высокое
  THEN клапан IS закрыт WITH 0.8;
  RULE 3 : IF температура IS горячая AND давление IS низкое
  THEN клапан IS закрыт;
  RULE 4 : IF температура IS горячая AND давление IS высокое
  THEN клапан IS открыт_на_наполнение;
```

```
END_RULEBLOCK
END_FUNCTION_BLOCK
```

Необязательные параметры (Optional parameters)

При построении нечетких моделей может возникнуть необходимость определить некоторую дополнительную информацию для той или иной системы, чтобы иметь возможность наиболее адекватного преобразования приложений нечеткого управления. Такая дополнительная информация может относиться к соответствующему элементу языка FCL и помещаться между ключевыми словами `OPTIONS` и `END_OPTIONS` следующим образом.

```
OPTIONS
  параметры_спецификации_приложения
END_OPTIONS
```

Соответствующие элементы языка FCL должны удовлетворять правилам согласованности классов открытого уровня (правило 6).

9.3. Пример разработки и записи нечетких моделей на языке FCL Нечеткая модель управления смесителем воды при принятии душа

Соответствующий пример был рассмотрен ранее в *лекции 8*. Если в качестве алгоритма нечеткого вывода используется алгоритм Мамдани, то методом активизации будет `MIN`, который рассчитывается по формуле (8.6). Для аккумуляции заключений правил будем использовать метод `MAX`, который наиболее часто применяется в случае схемы нечеткого вывода методом Мамдани. Наконец, в качестве метода дефазсификации будем использовать метод центра тяжести, который задается с помощью ключевого слова `COG`.

Таким образом, нечеткая модель управления смесителем воды может быть записана в нотации языка FCL следующим образом:

```
FUNCTION_BLOCK смеситель_воды
VAR_INPUT
  температура: REAL;
END_VAR
VAR_OUTPUT
  угол: REAL;
END_VAR
FUZZIFY температура
  TERM холодная := (10, 1) (30, 0);
  TERM прохладная := (20, 0) (35, 1) (50, 0);
  TERM теплая := (40, 0) (50, 1) (60, 0);
  TERM не_очень_горячая := (50, 0) (60, 1) (70, 0);
  TERM горячая := (60, 0) (70, 1);
END_FUZZIFY
DEFUZZIFY угол
  TERM большой_влево := (-72, 1) (-36, 0);
  TERM небольшой_влево := (-54, 0) (-27, 1) (0, 0);
  TERM нуль := (-18, 0) (0, 1) (18, 0);
  TERM небольшой_вправо := (0, 0) (27, 1) (54, 0);
  TERM большой_вправо := (36, 0) (72, 1);
  ACCU: MAX;
  METHOD : COG
  DEFAULT := 0
END_DEFUZZIFY
RULEBLOCK Номер_1
  AND : MIN;
  RULE 1: IF температура IS горячая THEN угол IS большой_вправо;
  RULE 2: IF температура IS не_очень_горячая THEN угол IS неболь-
  шой_вправо;
  RULE 3: IF температура IS теплая THEN угол IS нуль;
```

```
RULE 4: IF температура IS прохладная THEN угол IS небольшой_ влево"  
RULE 5: IF температура IS холодная THEN угол IS большой_ влево;  
END_RULEBLOCK  
END_FUNCTION_BLOCK
```

Литература:

Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. — СПб.: БХВ Петербург, 2005. — 736 с.: ил.