



Язык нечеткого управления

Деменков Н.П., МГТУ им. Н.Э. Баумана,
г. Москва, Россия

Язык нечеткого управления (Fuzzy Control Language - FCL) разработан для представления нечетких моделей систем управления, в частности, моделей ПЛК в форме структурируемого текста, который может быть интерпретирован как программа на языке высокого уровня. Язык описан в стандарте IEC 61131-7, в котором определяются цели разработки этого языка, его базовая нотация, и приводятся примеры записи моделей нечеткого управления с использованием нотации. Хотя стандарт не определяет требования к вычислительным устройствам и средам, которые могут реализовывать трансляцию, компиляцию и выполнение программ на FCL, описанная в нем нотация основных компонентов систем нечеткого вывода позволяет достичь формального уровня строгости, необходимого для последующей разработки соответствующих инструментальных средств.

Под нечетким управлением (Fuzzy Control) понимается область применения общей методологии теории нечетких множеств и нечеткой логики для решения практических задач управления. Нечеткое управление возникло как технология, способная расширить возможности автоматизации производства и предназначенная для решения прикладных задач в области управления, которые в общем случае могут быть реализованы с помощью ПЛК.

Нечеткое управление базируется на использовании не столько аналитических или теоретических моделей, сколько на практическом применении знаний, которые можно представить в форме баз правил, называемых лингвистическими. Нечеткое управление может использоваться в том случае, когда существует определенный опыт экспертов и его можно записать некоторым формальным образом. Широкий диапазон приложений и естественность подхода, основанного на

опыте специалистов, делает нечеткое управление основным средством, которое в качестве стандарта должно стать доступным для всех пользователей ПЛК. Нечеткое управление может также непосредственно комбинироваться с классическими методами управления.

Применение нечеткого управления может быть наиболее эффективным в тех случаях, когда отсутствует явная модель процесса, или аналитическая модель является слишком сложной для представления или получения решений в реальном масштабе времени.

Другое достоинство нечеткого управления заключается в непосредственном объединении опыта нескольких специалистов. Нечеткое управление, являясь многозначным управлением, больше ограничивается значениями высказываний "истина" или "ложь". Эта особенность делает нечеткое управление адекватным средством для моделирования эмпирического опыта экспертов, оперируя теми понятиями, в терминах которых формулируются управляющие воздействия на заданном множестве входов.

С точки зрения информационных технологий системы нечеткого управления являются производными экспертными системами. С точки зрения теории систем управления системы нечеткого управления являются нелинейными регуляторами. При этом текущие значения выходных переменных зависят только от текущих значений входных переменных и не зависят от предыстории этих значений, за исключением случаев, когда отсутствуют активные правила, и не определены значения переменных по умолчанию. Если же регулятор должен быть реализован как динамическая система, то соответствующие динамические функции представляют собой внешние элементы для нечеткого функционального блока.

Область применения нечеткого управления достаточно широка - от небольших и простых приложений до комплексных и сложных проектов. Чтобы охватить все возможные случаи, следует использовать правила согласованности классов систем нечеткого управления, которые дополняют и расширяют базовую нотацию FCL. При этом базовый класс определяет минимальное множество требований, которым должны удовлетворять все согласованные системы, что обеспечивает переносимость программ нечеткого управления.

Цель стандарта состоит в том, чтобы, с одной стороны, предоставить разработчикам и пользователям однозначные и доступные для понимания базовые средства для интеграции приложений нечеткого управления, которые могут быть записаны в нотации языков ПЛК, а с другой - обеспечить возможность переносимости мобильных программ нечеткого управления между различными системами программирования.

Необязательные средства FCL-определены в классе расширения. Программы нечеткого управления, использующие эти средства, могут переноситься с одной системы на другую только в том случае, если системы реализуют одинаковое множество этих средств. В противном случае может оказаться возможным лишь частичный перенос программ. Стандарт не требует, чтобы все согласованные системы реализовывали средства класса расширения в полном объеме. Хотя и допускается возможность частичного переноса, следует избегать использования нестандартных средств. Поэтому согласованная система не должна содержать нестандартные средства, которые не могут быть адекватно реализованы с использованием стандартных средств базового класса и класса расширения. Чтобы не исключать из рассмотрения

системы, использующие свои собственные и достаточно сложные средства, и не препятствовать процессу дальнейшего развития, стандарт разрешает использование дополнительных нестандартных средств, которые не вошли ни в базовый класс, ни в класс расширения. Однако такие средства должны быть перечислены стандартным способом, чтобы можно было легко установить их нестандартный характер.

Переносимость приложений нечеткого управления зависит от особенностей как систем программирования, так и от характеристик систем управления. Все эти особенности указываются в списке проверки данных, который разрабатывается производителями систем.

Приложения нечеткого управления, разработанные в форме программ на FCL, согласно стандарту IEC 61131-7, должны быть инкапсулированы в функциональные блоки (ФБ) на основе стандарта IEC 61131-3, который определяет использование языков программирования в ПЛК. Тип ФБ, определяемый в FCL, задается входными и выходными параметрами, специальными правилами и объявлениями нечеткого управления. Соответствующие экземпляры ФБ должны содержать значения данных конкретных приложений нечеткого управления. ФБ, записанные на FCL, могут быть использованы на любом из пяти языков стандарта IEC 61131-3. При этом типы данных входных и выходных параметров ФБ на FCL должны соответствовать аналогичным параметрам такой среды реализации.

Определение нотации FCL базируется на определениях языков программирования ПЛК. Взаимодействие алгоритма нечеткого управления со средой программирования должно быть скрыто от соответствующих программ. Именно поэтому алгоритм нечеткого управления следует представлять в форме ФБ. Для представления собственно лингвистических аспектов ФБ нечеткого управления необходимы такие элементы, как функции принадлежности, правила, операции и методы, определяемые стандартом. Стандартизация элементов FCL выполнена с целью достижения единого представления данных для обеспечения переноса программ между системами нечеткого управления различными производителями с помощью текстового файла на FCL в формате переноса данных. Использование единого представления разра-

ботчиками ПЛК позволяет каждому из них иметь собственное аппаратное обеспечение, редакторы и компиляторы программ. Разработчику достаточно соответствующих проектов нечеткого управления между различными разработчиками.

В соответствии со стандартом для FCL используется следующее подмножество элементов языка для текстовых языков программирования: буквы, цифры, идентификаторы; константы; типы данных; переменные, а для записи правил продукции

следующие элементы FCL: объявление функционального блока, входных и выходных переменных, других переменных, блоков фазификации, дефазификации, правил, необязательного блока, лингвистических термов, функции принадлежности, метода дефазификации, активации и аккумуляции, определение операций, условий и подусловий.

В табл. 1 приведен перечень ключевых слов FCL, которые разрешается использовать только в том значении, в котором оно определено в нотации языка.

Таблица 1

ключевое слово	описание
()	скобки в условии, терме, интервале
ACCU	метод аккумуляции
ACT	метод активизации
AND	операция логической конъюнкции (логическое И)
ASUM	операция логическое ИЛИ по формуле алгебраической суммы
BDIF	операция логическое И по формуле ограниченной разности
BSUM	метод аккумуляции по формуле ограниченной суммы
COA	метод дефазификации по формуле центра площади
COG	метод дефазификации по формуле центра тяжести
COGS	метод дефазификации по формуле центра тяжести для одноэлементных множеств
DEFAULT	значение выхода по умолчанию в случае отсутствия активных правил
DEFUZZIFY	дефазификация выходной переменной
END_DEFUZZIFY	конец спецификации дефазификации
END_FUNCTION_BLOCK	конец спецификации функционального блока
END_FUZZIFY	конец спецификации фазификации
END_OPTIONS	конец спецификации дополнительных параметров
END_RULEBLOCK	конец спецификации блока правил
END_VAR	конец определения входных/выходных переменных
FUNCTION_BLOCK	начало спецификации функционального блока
FUZZIFY	фазификация входной переменной
IF	начало правила, после которого следует его условие
IS	связка для значения лингвистической переменной в условии или заключении
LM	метод дефазификации по формуле левого модального значения
MAX	метод аккумуляции по формуле максимума
METHOD	метод дефазификации
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 дефазификации 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

Согласно стандарту внешнее представление нечеткого ФБ требует использования следующих элементов языка:

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 позволяют описать интерфейс ФБ (Function Block interface). Интерфейс ФБ определяется совместно с параметрами, которые могут использоваться внутри и вне этого ФБ. Типы данных этих параметров должны быть определены в соответствии с требованиями стандарта FCL.

Фазификация (Fuzzyfication)

в FCL описывается между ключевыми словами FUZZIFY и END_FUZZIFY следующим образом:

FUZZIFY имя_переменной

TERM имя_терма := функция_принадлежности;

END_FUZZIFY

После ключевого слова FUZZIFY должно быть записано имя переменной, которая используется для фазификации. Это имя переменной предварительно должно быть определено в секции VAR_INPUT. Соответствующая лингвистическая переменная должна быть описана одним или несколькими лингвистическими термами, которые определяются после ключевого слова TERM и описываются посредством функций принадлежности, фазифицируя эту переменную.

В качестве функции принадлежности может использоваться некоторая кусочнолинейная функция; она задается таблицей своих точек. 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. Точки должны записываться в возрастающем порядке значений переменной. Число точек может изменяться, но их максимальное количество ограниче-

но согласно правилу 6 классов согласованности. 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.

Дефазификация (Defuzzification). Лингвистическая переменная должна быть преобразована в количественное значение. Это преобразование описывается между ключевыми словами DEFUZZIFY и END_DEFUZZIFY следующим образом:

DEFUZZIFY имя переменной

TERM имя_терма := функция_принадлежности;

Метод_дефазификации;

Значение_по_умолчанию;

[интервал];

END_DEFUZZIFY.

Переменная, которая используется для дефазификации, должна быть записана в поле ключевого слова DEFUZZIFY. Имя этой переменной следует объявить ранее в секции VAR_OUTPUT. Определение лингвистических термов задается в секции фазификации. С целью упрощения дефазификации выходов могут быть использованы специальные функции принадлежности в форме значения для одноэлементного множества. Подобные функции принадлежности описываются посредством единственного значения для лингвистического терма. Метод дефазификации должен быть определен посредством элемента языка МЕТОД после ключевого слова METHOD.

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 дефазификация 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. 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. Интервал используется для спецификации минимального и максимального значений некоторой выходной переменной. Если функция 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 определить, 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.

Блок правил (Rule block). Для определения механизма нечеткого вывода используется один или несколько блоков правил. Для 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

Таблица2

ключевое слово	алгоритм	ключевое слово	алгоритм
MAX	$\max\{\mu 1(x), \mu 2(x)\}$	MIN	$\min\{\mu 1(x), \mu 2(x)\}$
ASUM	$\mu 1(x) + \mu 2(x) - \mu 1(x) \cdot \mu 2(x)$	PROD	$\mu 1(x) \cdot \mu 2(x)$
BSUM	$\min\{1, \mu 1(x) + \mu 2(x)\}$	BDIF	$\max\{0, \mu 1(x) + \mu 2(x) - 1\}$

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 **таблице 2.**

Метод активизации определяется после ключевого слова АСТ следующим образом:

АСТ : метод_активизации;

Методы активизации, которые допустимы в FCL, определены в **таблице 3.**

название метода	ключевое слово	алгоритм
произведение	PROD	$\mu_1(x) \cdot \mu_2(x)$
минимум	MIN	$\min\{\mu_1(x), \mu_2(x)\}$

При этом следует помнить, что метод активизации не применим для одноэлементного множества. Метод аккумуляции определяется после ключевого слова ACCU следующим образом:

ACCU : метод_аккумуляции.

Методы аккумуляции, которые

название метода	ключевое слово	алгоритм
максимум	MAX	$\max\{\mu_1(x), \mu_2(x)\}$
ограниченная сумма	BSUM	$\min\{1, \mu_1(x) + \mu_2(x)\}$
нормализованная сумма	NSUM	$\mu_1(x) + \mu_2(x) / \max\{1, \mu_1(y) + \mu_2(y)\}, y \in X$

допустимы в FCL, определены в **таблице 4.**

Входами блока правил являются лингвистические переменные со своими множествами лингвистических термов. Каждый из термов имеет соответствующие значения функции принадлежности. Правила определяются внутри блока. Каждое правило должно иметь в качестве своего имени уникальный номер внутри блока.

Этот номер записывается после ключевого слова RULE и заканчивается двоеточием. Са-

мо правило должно начинаться ключевым словом IF, после которого

записывается условие правила. После условия должно следовать заключение, которое записывается за ключевым словом 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 ФБ. 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 **таблицы 5.**

В соответствии с базовым уровнем согласованности операция OR в условии некоторого правила может

операция	приоритет
скобки ()	1
NOT (НЕ)	2
AND (И)	3
OR (ИЛИ)	4

быть заменена определением двух отдельных правил. 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. Допускается использование ключевого слова NOT перед записью подусловия. В этом случае следует применять круглые скобки.

Заключение может быть разделено на несколько подзаключений и

Передплатный индекс 23717

електро ТЕМА
всеукраїнська галузева газета

*Читайте!
Це Ваша газета.*

выходные переменные. Подзаключения начинаются с имени некоторой лингвистической переменной, после которого следует ключевое слово IS с одним лингвистическим термом для этой лингвистической переменной. 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. Что-бы 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 даст возможность изменять весовой коэффициент в ходе выполнения программы и тем самым адаптировать программу нечеткого управления к особенностям решаемой задачи. Если подзаключение не имеет записи с ключевым словом WITH, то используется значение весового коэффициента по умолчанию, которое равно 1.

Необязательные параметры (Optional parameters). При построении нечетких моделей может возникнуть необходимость определения некоторой дополнительной информации для той или иной системы, чтобы иметь возможность наиболее адекватного преобразования приложений нечеткого управления. Такая дополнительная информация может относиться к соответствующему элементу языка и помещаться между ключевыми словами OPTIONS и END_OPTIONS следующим образом:

OPTIONS

Параметры_спецификации_приложения
END_OPTIONS.

Соответствующие элементы языка должны удовлетворять правилам согласованности классов открытого уровня.

Согласованность классов FCL. Модели систем управления, записанные в нотации FCL, должны быть согласованными по так называемым уровням согласованности классов. Согласно стандарту уровни согласованности представляют собой иерархию из трех уровней:

- базовый, включающий определение ФБ и типы данных;
- расширения, включающий необязательные параметры;
- открытый, включающий дополнительные параметры.

Модель системы управления должна соответствовать требованиям стандарта, которые представлены в форме правил согласованности, приводимых ниже.

1. Для реализации функциональности в модели нечеткого управления следует использовать ФБ, как он определен в настоящем стандарте, поэтому определения функциональных блоков и типов данных для входных и выходных параметров ФБ должны согласовываться с базовой нотацией языка.

2. Все параметры функциональности модели нечеткого управления,

в том случае все модели стандартных систем.

Подмножество элементов уровня расширения, определенных в таблице 7, составляет дополнительные элементы, которые можно использовать в необязательном порядке; все необязательные параметры должны относиться к уровню расширения, а перечень используемых необязательных параметров в форме **таблицы 7** должен стать частью документации на модель системы.

4. Другие параметры, выходящие за пределы базового уровня и уровня расширения, допускается ис-

Таблица 6

элемент FCL	ключевое слово	примечание
объявление функционального блока	VAR_INPUT VAR_OUTPUT	содержит входные и выходные переменные
функция принадлежности	входная переменная: TERM выходная переменная: TERM	максимум 3 точки (координата степени принадлежности равна 0 или 1) только одноэлементное множество
агрегирование условий	оператор: AND	алгоритм MIN
активизация	AKT	не применяется в случае использования только одноэлементных множеств
аккумуляция (результат агрегирования)	оператор: ACCU	алгоритм: MAX
дефазификация	METHOD	алгоритм: COGS
значение по умолчанию	DEFAULT	чтобы использовать этот параметр, в базовом классе должны быть подготовлены предварительные определения
условие	IFO ISO	N подусловий
заклучение	THEN	только одно подзаклучение
весовой коэффициент	WITH	только значение

определенные в **таблице 6**, должны быть заданы в соответствии с базовой нотацией языка; табл. 6 определяет множество элементов базового уровня, которые должны содержать в об-

пользовать в модели таким образом, чтобы эти параметры не имели таких же или аналогичных параметров среди стандартных элементов функциональности FCL. Тем самым должны

Таблица 7

элемент FCL	ключевое слово	примечание
объявление функционального блока	VAR	содержит локальные переменные
функция принадлежности	входная переменная: TERM выходная переменная: TERM	максимум 4 точки (координата степени принадлежности равна 0 или 1)
агрегирование условий	оператор: AND оператор: OR оператор: NOT скобки	алгоритм: PROD, BDIF алгоритм: ASUM, BSUM 1 - {аргумент} ()
активизация	оператор: ACT	алгоритм: MIN, PROD
аккумуляция	оператор: ACCU	алгоритм: BSUM, NSUM
метод дефазификации	оператор: METHOD	алгоритм: COG, COA, LM, RM
значение по умолчанию	DEFAULT	NC, значение
условие	IF	n подусловий, N входных переменных
заклучение	THEN	n подзаклучений, N выходных переменных
весовой коэффициент	WITH	значение назначается переменной в секции объявления VAR_INPUTO VAR_END

быть исключены любые возможные недоразумения. Эти дополнительные параметры относятся к открытому уровню, а перечень используемых дополнительных параметров в форме **таблицы 8** должен стать частью документации на модель системы.

5. Перенос прикладных программ между различными системами нечеткого управления следует осуществлять в текстовой форме в соответствии с нотацией языка.

Список проверки данных. Этот список должен прилагаться к технической документации. В этом списке разработчик ПЛК, средств нечеткого программирования и прикладных программ должен описать специальные параметры реализации своей системы нечеткого управления. Чтобы обеспечить передачу приложений нечеткого управления между системами различных производителей, необходимо представить список проверки данных (**табл. 9**), который является средством установления правильности возможного переноса программ.

В качестве примера рассмотрим нечеткую модель управления контейнерным краном на языке FCL. Контейнерный кран используется при выполнении погрузочно-разгрузочных работ в портах для транспортировки моноблочного контейнера, с которым он соединяется гибким тросом. Кран поднимает контейнер к кабине крана, которая вместе с контейнером может перемещаться в горизонтальном направлении по направляющим. Когда контейнер поднимается к кабине, а кран приходит в движение, контейнер начинает раскачиваться и отклоняться от строго вертикального положения под кабиной крана.

Таблица 8

произвольные функции принадлежности для входов/выходов (например, функция Гаусса, экспоненциальная функция)
более чем 4 точки для функции принадлежности
координата значения функции принадлежности от 0 до 1
произвольные элементы, заданные программистом

Проблема заключается в том, что пока контейнер раскачивается в ходе своей транспортировки и откло-

няется от вертикали, он не может быть опущен на основании цели перемещения, в качестве которой используются железнодорожные платформы или другие транспортные средства. Чтобы иметь возможность автоматически управлять таким краном, необходимо использовать дат-

Таблица 9

технические данные	значения, установленные разработчиком
типы данных входов и выходов ФБ	REAL, INT
строки комментариев в программах на FCL	YES, NO
длина идентификаторов	
(например, имена переменных, блоков правил, термов)	6, 8
максимальное число входных переменных для фазификации	6, 8
максимальное число функций принадлежности	
термов для входной переменной	5,7
максимальное общее число функций принадлежности	
термов для всех входных переменных	30,56
максимальное число точек функции принадлежности	
для каждого из термов входных переменных	3,4,10
максимальное общее число точек функции принадлежности	
для всех термов входных переменных	90, 224
максимальное число выходных переменных для дефазификации	6,8
максимальное общее число функций принадлежности термов	
для всех выходных переменных	30, 56
максимальное число точек функции принадлежности	
для каждого из термов выходных переменных	1, 4, 10
максимальное общее число точек функции принадлежности	
для всех термов выходных переменных	90, 224
максимальное число блоков правил	1, 10
максимальное число правил в блоке	10
максимальное число подусловий в правиле	4, 10
максимальное число всех правил	15
максимальное число подзаключений в правиле	4
глубина вложенности скобок ()	1, 3



чики для измерения горизонтального положения кабины крана ("расстояние") и измерения угла раскачивания контейнера ("угол"). Выходом в этом случае является мощность мотора. Анализ действий крановщиков-операторов, выполняющих управление краном, показывает, что они в своей работе применяют следующие эвристические правила:

- начинать движение следует со средней мощностью двигателя;

- если движение уже началось и кабина располагается далеко от цели, необходимо отрегулировать мощность двигателя таким образом, чтобы контейнер находился несколько впереди кабины крана;

- если кабина находится близко над целью, уменьшить ее скорость таким образом, чтобы контейнер нахо-

дился несколько впереди кабины крана;

- когда контейнер находится близко от позиции цели, нужно выключить мощность двигателя;

- когда контейнер находится прямо над позицией цели, следует остановить двигатель.

Для построения базы правил преобразуем эти 5 эвристических правил в 6 правил нечеткой продукции.

ПРАВИЛО_1: ЕСЛИ "расстояние далекое" И "угол равен нулю", ТО "мощность положительная средняя".

{Фото судового крана с контейнером}

ПРАВИЛО_2: ЕСЛИ "расстояние далекое" И "угол отрицательный малый", ТО "мощность положительная большая".

ПРАВИЛО_3: ЕСЛИ "расстояние далекое" И "угол отрицательный большой", ТО "мощность положительная средняя".

ПРАВИЛО_4: ЕСЛИ "расстояние среднее" И "угол отрицательный малый", ТО "мощность отрицательная средняя".

ПРАВИЛО_5: ЕСЛИ "расстояние близкое" И "угол положительный малый", ТО "мощность положительная средняя".

ПРАВИЛО_6: ЕСЛИ "расстояние ноль" И "угол равен нулю", ТО "мощность равна нулю".

Для входных лингвистических переменных "расстояние" и "угол" будем использовать 5 термов с кусочно-линейными функциями принадлежности. Мощность мотора включает также 5 термов, но с функцией принадлежности для одноточечных множеств.

В качестве метода агрегирования подусловий будем использовать операцию min-конъюнкции, поскольку во всех правилах 1-6 в качестве логической связки для подусловий применяется только нечеткая конъюнкция (операция "И"). В качестве схемы не-

четкого вывода будем использовать метод Мамдани, поэтому методом активизации будет MIN.

Для аккумуляции заключений правил будем использовать метод max-дизъюнкции, который также применяется в случае схемы нечеткого вывода методом Мамдани. В качестве метода дефазификации будем использовать метод центра тяжести для одноэлементных множеств COGS.

Нечеткая модель управления контейнерным краном может быть записана в нотации FCL следующим образом:

```

FUNCTION_BLOCK контейнерный_кран
VAR_INPUT
    расстояние: REAL;
    угол: REAL;
END_VAR
VAR_OUTPUT
    мощность: REAL;
END_VAR
asucontrol.ru
FUZZIFY расстояние
    TERM очень_далекое: = (-5,1) (0,0);
    TERM нуль:= (-5,0) (0,1) (5,0);
    TERM близкое:= (0,0) (5,1) (10,0);
    TERM среднее:= (5,0) (10,1) (22,0);
    TERM далекое:= (10,0) (22,1);
END_FUZZIFY
FUZZIFY угол
    TERM отр_большой:= (-50,1) (-5,0);
    TERM отр_малый:= (-50,0) (-5,1) (0,0);
    TERM нуль:= (-5,0) (0,1) (5,0);
    TERM пол_малый:= (0,0) (5,1) (50,0);
    TERM пол_большой:= (5,0) (50,1);
END_FUZZIFY
DEFUZZIFY мощность

```

```

METHOD: COGS
    DEFAULT:=0
    END_DEFUZZIFY

RULEBLOCK Номер_1
    AND: MIN;
    RULE 1: IF расстояние IS далекое
    AND угол IS ноль
    THEN мощность IS пол_средняя;
    RULE 2: IF расстояние IS далекое
    AND угол IS отр_малый
    THEN мощность IS пол_большая;
    RULE 3: IF расстояние IS далекое
    AND угол IS отр_большой
    THEN мощность IS пол_средняя;
    RULE 4: IF расстояние IS среднее
    AND угол IS отр_малый
    THEN мощность IS отр_средняя;
    RULE 5: IF расстояние IS близкое
    AND угол IS пол_малый
    THEN мощность IS пол_средняя;
    RULE 6: IF расстояние IS ноль
    AND угол IS ноль
    THEN мощность IS ноль;
    END_RULEBLOCK
END_FUNCTION_BLOCK

```

Поскольку этот ФБ записан в нотации FCL, который является языком программирования высокого уровня, то рассмотренная программа по своей структуре полностью соответствует некоторой процедуре. Формальными параметрами такой процедуры являются входные переменные блока правил, а возвращаемыми значениями - значения выходной переменной после дефазификации. Вызов такой процедуры в некоторой внешней программе может быть реализован, например, следующим образом:

```

контейнерный_кран (расстояние:= 12, угол:= 15);
VarP:= контейнерный_кран.мощность.

```

В этом случае переменной с именем VarP присваивается значение нечеткого вывода для блока правил, определенного в ФБ контейнерный_кран, при конкретных значениях входных переменных: 12 м для расстояния и 15° для угла.

Статья перепечатывается с разрешения редакции журнала "Промышленные АСУ и контроллеры", г.Москва



```

TERM отр_высокая:= -27;
TERM отр_средняя:= -12;
TERM нуль:= 0;
TERM пол_средняя:= 12;
TERM пол_высокая:= 27;
ACCU: MAX;

```

КОНТАКТЫ:
 т. (107-495) 263-6727
 e-mail: demenkov@iu1.bmstu.ru