

doi: 10.17586/2226-1494-2023-23-5-1041-1049

УДК 681.5:004.4

Среда визуальной разработки программ многомерных нечетких интервально-логических регуляторов

Андрей Федорович Антипин¹✉, Евгения Викторовна Антипина²

¹ Стерлитамакский филиал Уфимского университета науки и технологий, Стерлитамак, 453103, Российская Федерация

² Уфимский университет науки и технологий, Уфа, 450076, Российская Федерация

¹ andrejantipin@ya.ru ✉, <https://orcid.org/0000-0002-9151-4167>

² stepashinaev@ya.ru, <https://orcid.org/0000-0002-8458-9638>

Аннотация

Введение. Рассмотрен подход к синтезу и разработке многомерных нечетких интервально-логических регуляторов в созданной среде визуального программирования, генерирующей программный код регуляторов для контроллеров Simatic S7-300 и S7-400. **Метод.** Многомерные интервально-логические регуляторы используют математический аппарат нечетких регуляторов со смешанной (трапециевидной и прямоугольной) функцией принадлежности μ , равной единице. Предложенная модификация регуляторов является авторской разработкой и предназначена для использования в системах управления с достаточно большим объемом производственных правил, описывающих логику сложного технологического объекта без адекватной математической модели. Программное обеспечение среды визуальной разработки учитывает требования действующих редакций стандартов ГОСТ Р МЭК 61131-3 и 61131-7. **Основные результаты.** Представлен базовый вариант блок-схемы многомерного интервально-логического регулятора. Приведены схемы интерпретации непрерывных физических величин эквивалентной совокупностью термов. Сформулирован пропорциональный алгоритм деинтервализации. Показано, что синтез и программирование данной модификации нечетких регуляторов заключается в определении векторов термов входных и выходных непрерывных величин, а также взаимных связей между ними, составлении системы производственных правил и выборе соответствующего алгоритма деинтервализации. Описан подход к разработке многомерных интервально-логических регуляторов при помощи специальных программных средств (редакторов и приложений), входящих в состав среды визуальной разработки. **Обсуждение.** Предложенный подход обеспечивает наглядное и быстрое создание функциональных блоков регуляторов для программируемых логических контроллеров Simatic S7-300 и S7-400. При этом имеется возможность анализа и симуляции работы системы производственных правил и регуляторов в целом, что позволяет сократить сроки разработки и отладки систем управления на их основе.

Ключевые слова

нечеткий регулятор, среда визуальной разработки, редакторы, система производственных правил, программное обеспечение

Благодарности

Исследование выполнено в рамках государственного задания Министерства науки и высшего образования Российской Федерации (код научной темы FZWU-2023-0002).

Ссылка для цитирования: Антипин А.Ф., Антипина Е.В. Среда визуальной разработки программ многомерных нечетких интервально-логических регуляторов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 5. С. 1041–1049. doi: 10.17586/2226-1494-2023-23-5-1041-1049

Visual programming environment for multidimensional fuzzy interval-logic regulators

Andrey F. Antipin¹✉, Evgenia V. Antipina²

¹ Sterlitamak branch of Ufa University of Science and Technology, Sterlitamak, 453103, Russian Federation

² Ufa University of Science and Technology, Ufa, 450076, Russian Federation

¹ andrejantipin@ya.ru✉, <https://orcid.org/0000-0002-9151-4167>

² stepashinaev@ya.ru, <https://orcid.org/0000-0002-8458-9638>

Abstract

The approach to synthesis and development of multidimensional fuzzy interval-logic regulators in the created visual programming environment that generates the program code of regulators for Simatic S7-300 and S7-400 controllers is considered. Multidimensional interval-logic regulators use the mathematical apparatus of fuzzy regulators with mixed (trapezoidal and rectangular) membership function μ equal to one. The proposed modification of regulators is the author's development and is intended for use in control systems with a sufficiently large volume of product rules describing the logic of a complex technological object without an adequate mathematical model. The software of the visual development environment takes into account the requirements of the current editions of IEC 61131-3 and 61131-7 standards. The basic variant of the block diagram of a multidimensional interval-logic regulator is presented. Schemes for the interpretation of continuous physical quantities by an equivalent set of terms are given. The proportional deintervalization algorithm is formulated. It is shown that the synthesis and programming of this modification of fuzzy regulators consists in determining the vectors of terms of input and output continuous variables, as well as the mutual relations between them, compiling a system of production rules and selecting the appropriate algorithm of deintervalization. The approach to the development of multidimensional interval-logic regulators by means of special software tools (editors and applications) included in the visual development environment is described. The proposed approach provides a clear and quick creation of controller functional blocks for Simatic S7-300 and S7-400 programmable logic controllers. At the same time, it is possible to analyze and simulate the work of the system of production rules and regulators as a whole, which reduces the time of development and debugging of control systems based on them.

Keywords

fuzzy regulator, visual development environment, editors, production rule system, software

Acknowledgements

This research was funded by the Ministry of Science and Higher Education of the Russian Federation (scientific code FZU-2023-0002).

For citation: Antipin A.F., Antipina E.V. Visual programming environment for multidimensional fuzzy interval-logic regulators. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 5, pp. 1041–1049 (in Russian). doi: 10.17586/2226-1494-2023-23-5-1041-1049

Введение

В настоящее время многомерные нечеткие регуляторы все больше используются в качестве самостоятельных структурных звеньев при программировании контуров регулирования автоматизированных систем управления технологическими процессами. В то же время сложность разработки и настройки самих регуляторов при их интеграции в систему управления приводит к появлению гибридных решений, основанных на комбинации классических регуляторов: пропорционального, пропорционально-интегрального, пропорционально-интегрально-дифференциального (ПИД) и программной надстройки в виде дополнительных продукционных правил. Часто это связано с определением подходящих функций принадлежности непрерывных физических величин, а также алгоритмов их фаззификации и дефаззификации, что напрямую влияет на погрешность получаемых выходных значений и, как следствие, качество регулирования. Все это усложняет оптимизацию, оценку и анализ семантики программного кода регуляторов на предмет выявления ошибок, достаточности и избыточности правил.

Отметим, что в связи с отсутствием в современных программных средах разработки программ для программируемых логических контроллеров возможностей реализации нестандартных решений, связанных с

нечеткими регуляторами и их модификациями, сложность разработки программ возрастает многократно. В большинстве случаев это приводит к исключению из процесса программирования удобных для использования графических языков, определенных стандартом ГОСТ Р МЭК 61131-3-2016¹, оставляя на выбор только один из текстовых языков.

Проблемам разработки автоматизированных систем управления с использованием нечетких регуляторов посвящен ряд работ отечественных и зарубежных авторов. Так, в работе [1] отмечена высокая сложность алгоритмов нечеткого регулирования по сравнению с традиционными алгоритмами. В [2] предложено решение задачи оценки точности системы автоматического управления с нечетким ПИД-регулятором, а в [3] описана реализация нечеткой экспертной системы. В работах [4–7] для синтеза нечетких регуляторов использовано стороннее, по отношению к контроллерам, программное обеспечение MATLAB. Работа [8] посвящена описанию языка Fuzzy Applied Cell Control Technology и программного обеспечения на его основе для интеллектуальных систем управления. Трудности

¹ ГОСТ Р МЭК 61131-3-2016 | НАЦИОНАЛЬНЫЕ СТАНДАРТЫ [Электронный ресурс]. Режим доступа: <https://protect.gost.ru/v.aspx?control=8&id=195479>, свободный (дата обращения: 05.09.2023).

практической реализации нечетких регуляторов и систем представлены в работах [9–14].

В результате для синтеза нечетких регуляторов и разработки систем управления на их основе часто используется стороннее программное обеспечение, что приводит к дополнительным затратам и оказывает влияние на сроки разработки. При эксплуатации и обслуживании подобных систем собственными силами организаций обычно закрыт доступ к внутреннему содержанию блоков нечетких регуляторов (так называемое «ноу-хау» разработчиков).

Как следствие, актуальным становится внедрение новых программных решений в виде специальных редакторов в состав известных сред разработки программ программируемых логических контроллеров, что является наиболее предпочтительным вариантом, а также разработка программного обеспечения, позволяющего решать узконаправленные задачи, с возможностью его интеграции с общепризнанными мировыми продуктами. Последний способ влечет за собой ряд дополнительных трудностей, связанных с изучением форматов файлов известных систем и сред разработки (при условии, что они являются открытыми и общедоступными) и возможностей интеграции, что в подавляющем большинстве случаев возможно только при выполнении первого условия.

Научная новизна настоящей работы заключается в создании подхода к синтезу и разработке программного кода многомерных нечетких регуляторов (на примере отдельно взятой их модификации). В работе использована созданная среда визуальной разработки, генерирующая программный код в формате структурированного языка программирования SCL для контроллеров Simatic S7-300 и Simatic S7-400 с учетом ГОСТ Р МЭК 61131-7-2017¹.

Многомерный интервально-логический регулятор

Многомерные интервально-логические регуляторы (МИЛР) представляют собой частный случай нечетких регуляторов со смешанной (трапецевидной и прямоугольной) функцией принадлежности μ , равной единице. Данная модификация регуляторов является новой разработкой и предназначена для использования в системах управления с большим объемом продукционных правил, описывающих логику работы сложного технологического объекта без адекватной математической модели.

На рис. 1 представлен базовый вариант блок-схемы МИЛР, где I_1, \dots, I_n и D_1^y, \dots, D_k^y — интерваллизаторы и деинтерваллизаторы входных и выходных непрерывных величин x_1, \dots, x_n и y_1, \dots, y_k ; D_1^z, \dots, D_m^z — деинтерваллизаторы выходных непрерывных величин z_1, \dots, z_m с обратной связью; БПП — блок продукционных правил МИЛР; ID — блок формирования идентификационного

номера составного продукционного правила, описывающего ситуацию в текущий момент времени t ; БИ и БД — условные блоки интервализации и деинтервализации, состоящие из интерваллизаторов I_1, \dots, I_n и деинтерваллизаторов D_1^y, \dots, D_k^y и D_1^z, \dots, D_m^z МИЛР; БРФ МИЛР — блоки расширения функций МИЛР (необязательно), такие как блок предсказания термов, блок оптимизации, блок тревоги и др.; T_t — порядковые номера термов переменных МИЛР в момент времени t ; БЛВ — блок логического вывода.

В МИЛР процедуры фаззификации и дефаззификации упрощены до определения принадлежности значений непрерывных величин тем или иным термам, которые, в свою очередь, определяют некоторые непересекающиеся диапазоны (или интервалы) значений. В связи с этим они заменены на наиболее подходящие по смыслу понятия интервализации и деинтервализации.

Схемы интервализации непрерывной физической величины x эквивалентной совокупностью n термов T приведены на рис. 2. Левое ($x^{(n)}$) и правое ($x^{(n)}$) значения определяют диапазоны значений термов, что позволяет задать характер определения принадлежности термов T и погрешность вычислений.

Например, при прямом характере определения принадлежности терм-множество T^x (рис. 2, а) аналитически можно представить:

$$T^x = \{T_1^x(x_1^{(n)} \leq x_t < x_1^{(n)}), T_2^x(x_2^{(n)} \leq x_t < x_2^{(n)}), \dots, T_n^x(x_n^{(n)} \leq x_t \leq x_n^{(n)})\},$$

а при обратном характере в виде:

$$T^x = \{T_1^x(x_1^{(n)} \leq x_t \leq x_1^{(n)}), T_2^x(x_2^{(n)} < x_t < x_2^{(n)}), \dots, T_n^x(x_n^{(n)} < x_t \leq x_n^{(n)})\},$$

где x_t — значение непрерывной величины x в момент времени t .

Ширина термов определяет погрешность при интервализации и деинтервализации значений непре-

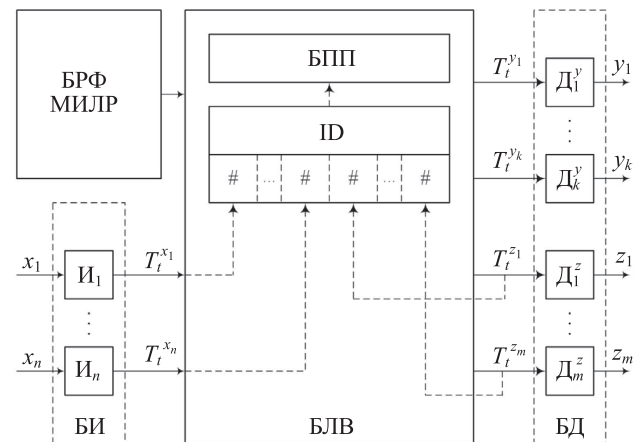


Рис. 1. Базовый вариант блок-схемы многомерного интервально-логического регулятора

Fig. 1. Basic version of the block diagram of multidimensional interval-logic regulator

¹ ГОСТ Р МЭК 61131-7-2017 | НАЦИОНАЛЬНЫЕ СТАНДАРТЫ [Электронный ресурс]. Режим доступа: <https://protect.gost.ru/v.aspx?control=8&id=210690>, свободный (дата обращения: 05.09.2023).

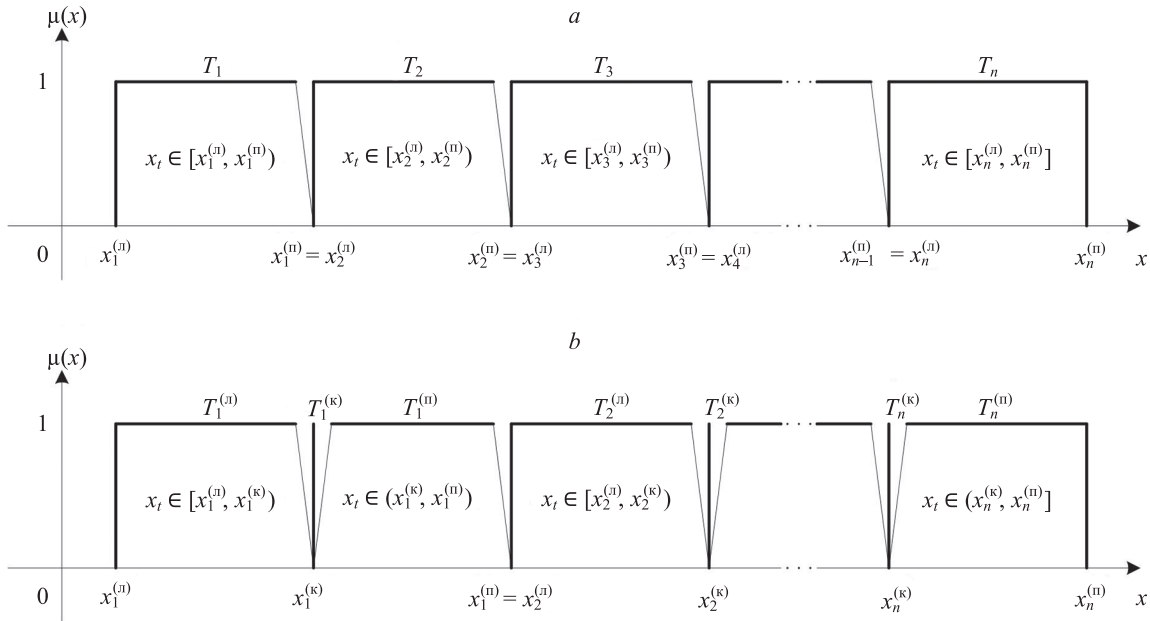


Рис. 2. Схемы интервализации непрерывной физической величины x эквивалентной совокупностью термов T : с прямым характером определения принадлежности термов (а); с контрольными четкими значениями, определяющими термы (б)

Fig. 2. Schemes of intervalization of a continuous physical quantity x by an equivalent set of terms T : with direct character of determining the membership of terms (a); with control values defining terms (b)

рывных величин МИЛР. Для более гибкой настройки величины погрешности возможен следующий вариант интерпретации непрерывных величин совокупностью термов, при котором каждый терм задается тройкой $(x^{(k)}, x^{(n)}, x^{(n)})$, где $x^{(k)}$ — конкретное четкое значение или контрольное значение, определяющее терм. При этом $x^{(k)} \in [x^{(n)}, x^{(n)}]$.

В зависимости от характера определения принадлежности термов МИЛР термы T непрерывной величины x задаются следующим образом:

$$\begin{aligned} &T_i^{(k)}, \text{ если } x_t = x_i^{(k)}, i \text{ — любое;} \\ &T_i^{(n)}, \text{ если } x_t \in [x_i^{(n)}, x_i^{(k)}], i \text{ — любое;} \\ &T_i^{(n)}, \text{ если } x_t \in (x_i^{(k)}, x_i^{(n)}), i < n; \\ &T_i^{(n)}, \text{ если } x_t \in (x_i^{(k)}, x_i^{(n)}], i = n, \end{aligned}$$

или

$$\begin{aligned} &T_i^{(k)}, \text{ если } x_t = x_i^{(k)}, i \text{ — любое;} \\ &T_i^{(n)}, \text{ если } x_t \in [x_i^{(n)}, x_i^{(k)}], i = 1; \\ &T_i^{(n)}, \text{ если } x_t \in (x_i^{(n)}, x_i^{(k)}], i > 1; \\ &T_i^{(n)}, \text{ если } x_t \in (x_i^{(k)}, x_i^{(n)}], i \text{ — любое,} \end{aligned}$$

где $i = 1, 2, \dots, n$.

Пример схемы интервализации непрерывной величины x МИЛР эквивалентной совокупностью термов T по описанному способу приведен на рис. 2, б.

Максимальное число продукционных правил МИЛР R_{\max} , составляющих систему продукционных правил (СПП), в объеме, зависящем от суммарного числа термов K непрерывных величин МИЛР, может быть рассчитано по формуле

$$R_{\max} = \prod_{i=1}^n K(x_i) \prod_{j=1}^m K(z_j).$$

Каждое продукционное правило БПП однозначно определяет вектор термов входных и выходных величин МИЛР и имеет вид:

$$T_t^{y,z} = F(T_t^{x_1}, \dots, T_t^{x_{ns}}, T_t^{z_1}, \dots, T_t^{z_{ms}}),$$

где F — некоторая логическая функция, определяющая взаимную связь выходной величины с ns и ms входными величинами x и z соответственно.

Термы $T_t^{y,z}$ выходных непрерывных величин y, z равны логической единице, если связанные с ними термы входных величин также равны логической единице.

Графически продукционное правило для выходной величины y_k можно изобразить в виде, представленном на рис. 3, где штрихом обозначены термы, значение которых равно логической единице.

Идентификационный (или ID) номер, формируемый из совокупности порядковых номеров термов, полученных при опросе входов в начале рабочего цикла контроллера, однозначно определяет набор управляющих воздействий МИЛР в текущий момент времени t .

Порядковые номера термов выходных величин подаются в блок деинтервализации, где деинтервалитаторы производят расчет четких значений, подаваемых на выходы МИЛР, по заранее определенному алгоритму.

Для деинтервализации значений выходных непрерывных величин МИЛР авторами разработаны соответствующие алгоритмы [15], из которых наиболее предпочтительным является пропорциональный алгоритм деинтервализации, что связано, прежде всего, с его простотой и применимостью в подавляющем большинстве случаев.

Так, значение выходной непрерывной величины y в границах $y_t^{(n)}, y_t^{(n)}$ ее рабочего терма T_t^y в момент

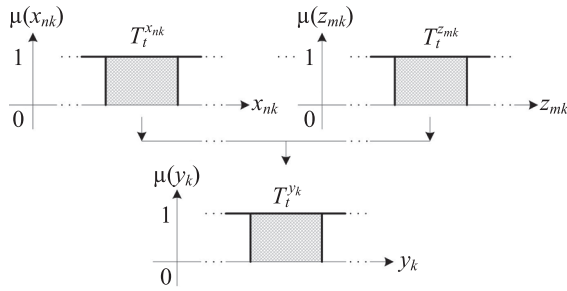


Рис. 3. Вариант графического представления
производственного правила

Fig. 3. A variant of the graphical representation of the
production rule

времени t может быть рассчитано пропорционально значениям связанным с ней ns и ms входными величинами x, z соответственно в рамках границ $x_t^{(n)}, x_t^{(n)}$ и $z_t^{(n)}, z_t^{(n)}$ их рабочих термов по формуле:

$$y = y_t^{(n)} + k_t^y (y_t^{(n)} - y_t^{(n)}),$$

где k_t^y — коэффициент пропорциональности, рассчитываемый согласно выражениям:

$$k_t^y = \frac{\sum_{i=1}^{ns} k_t^{x_i} + \sum_{j=1}^{ms} k_t^{z_j}}{ns + ms}, k_t^{x_i} = \frac{x_{it} - x_{it}^{(n)}}{x_{it}^{(n)} - x_{it}^{(n)}}, x_{it}^{(n)} \neq x_{it}^{(n)},$$

$$k_t^{z_j} = \frac{z_{jt} - z_{jt}^{(n)}}{z_{jt}^{(n)} - z_{jt}^{(n)}}, z_{jt}^{(n)} \neq z_{jt}^{(n)}.$$

Если зависимости между входными и выходными величинами МИЛР имеют более сложный вид, выражение для расчета коэффициента пропорциональности k_t^y может быть дополнено с целью учета влияния дополнительных факторов, таких как вид зависимости (прямой или обратный), степень (или коэффициент) влияния и пр.

Следовательно, синтез и программирование базовых функций МИЛР сводится к определению векторов термов входных и выходных непрерывных величин, а также взаимных связей между ними, составлению СПП и выбору соответствующего алгоритма деинтервализации.

Принцип работы в среде визуальной разработки МИЛР

Созданная среда визуальной разработки программ МИЛР представляет собой комплект редакторов и приложений [15, 16], предназначенных для быстрой разработки программ МИЛР, расчета специфических характеристик и анализа СПП.

Вид базового редактора программ МИЛР показан на рис. 4. Базовый редактор предназначен для визуального программирования функциональных блоков МИЛР, которые впоследствии могут быть переведены в формат языков программирования известных сред разработки программ промышленных контроллеров. Данный подход упрощает процесс разработки МИЛР (о чем было сказано ранее в разделе «Многомерный интервально-логический регулятор»).

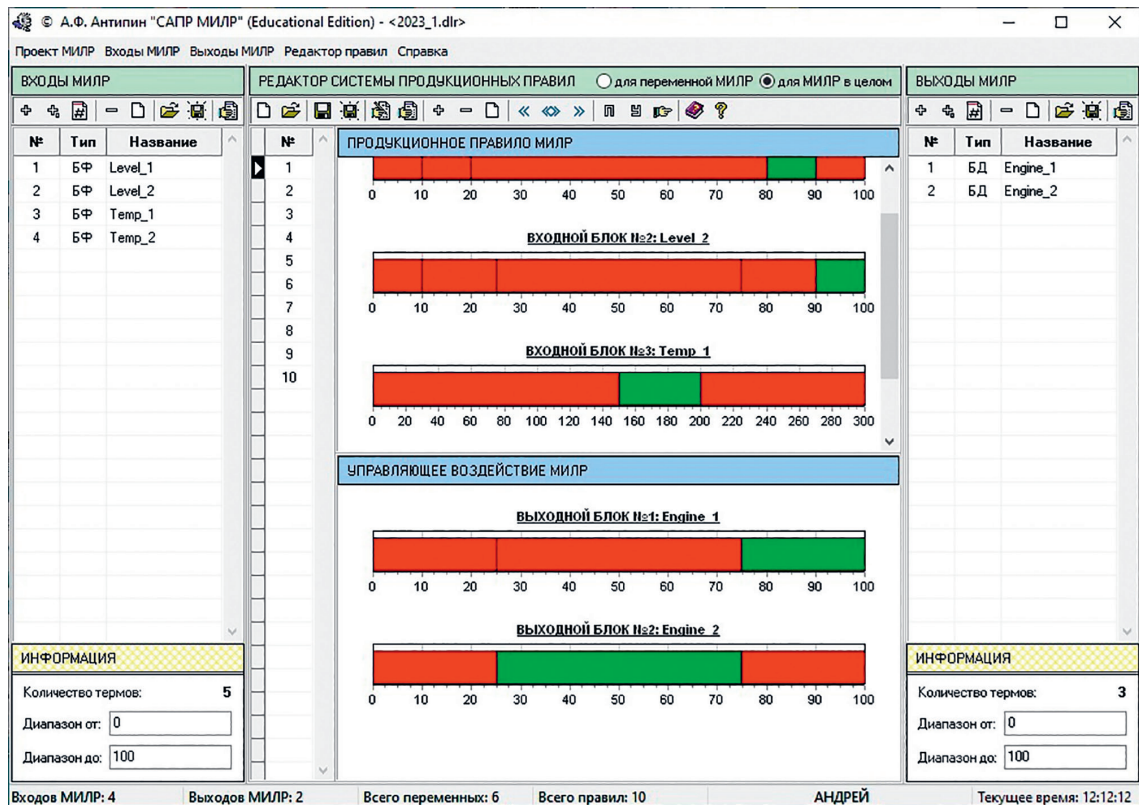


Рис. 4. Базовый редактор программы многомерного интервально-логического регулятора

Fig. 4. Basic editor of the program of multidimensional interval-logic regulator

Редактор программ МИЛР условно разделен на три части. В левой и правой частях окна редактора размещены панели для управления работой интервализаторов и деинтервализаторов с указанием терм-разбивки входных и выходных непрерывных величин. Также возможен выбор характера определения принадлежности термов (прямой или обратный).

В центре окна расположен редактор СПП МИЛР, который разделен на две части для раздельной работы с термами входных и выходных величин (верхняя и нижняя панели редактора). Окно редактора содержит фрагменты координатных осей, определенные диапазоном значений непрерывных величин МИЛР и их терм-разбивкой. Создание нового производственного правила состоит в указании термов входных величин, инициирующих его выполнение, а также термов выходных величин для определения значений управляющих воздействий в текущей ситуации.

По окончании работы с редактором возможен экспорт готовой программы МИЛР или отдельных функциональных блоков (интервализаторов и деинтервализации) в формат языков программирования SCL и AWL для программируемых логических контроллеров Simatic S7.

Для анализа итоговой совокупности переменных, а также СПП МИЛР существует отдельный редактор, внешний вид которого представлен на рис. 5. Данный редактор полностью совместим с проектами программ МИЛР, разработанных с помощью базового редактора, и позволяет дополнять их.

Процесс подготовки программы МИЛР к анализу состоит из следующих шагов, некоторые из которых можно пропустить при открытии базового проекта.

Шаг 1. Определение общего числа переменных или непрерывных величин МИЛР.

Шаг 2. Настройка переменных МИЛР путем задания имени и типа переменной, определения приоритета

(высокий или нормальный) для выделения критически важных производственных правил в общем объеме правил, входящих в СПП, диапазона терм-разбивки и суммарного количества термов. В редакторе определено три типа переменных МИЛР: IN (входная), OUT (выходная) и IN/OUT (выходная с обратной связью).

Шаг 3. Определение диапазонов термов непрерывных величин, включая установку принадлежности границ к тому или иному терму. Термы могут быть помечены специальным флагом «критический» для указания особой важности (например, при определении крайних диапазонов значений непрерывных величин), что позволяет выделить особо важные производственные правила.

Шаг 4. Настройка взаимных связей переменных МИЛР (о необходимости и важности которой говорится в работе [16]), т. е. определение функциональных зависимостей между входными и выходными величинами.

На рис. 5, б показано окно с результатами анализа переменных и СПП МИЛР. Таблица «Анализ переменных» содержит краткую информацию по переменным МИЛР: общее количество термов и количество критических термов (для каждой переменной и МИЛР в целом), отношение переменных к конкретной группе и др.

Таблица «Анализ правил СПП» содержит значения параметров МИЛР, зависящих от содержимого СПП, из которых можно выделить следующие.

1) **Максимальное R_{max} и фактическое $R_{факт}$ количество производственных правил, образующих СПП.** Данные параметры позволяют определить количество отсутствующих правил, необходимых для корректной работы МИЛР.

2) **Количество групп взаимосвязанных переменных G и суммарное число производственных правил для каждой группы.** Данные параметры характеризуют взаимосвязь входных и выходных величин и их принадлежность к той или иной условной группе. Кроме общих групп, показаны критические группы, содер-

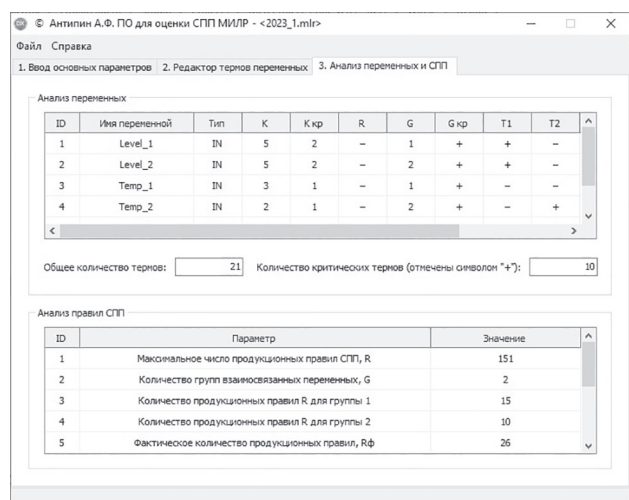
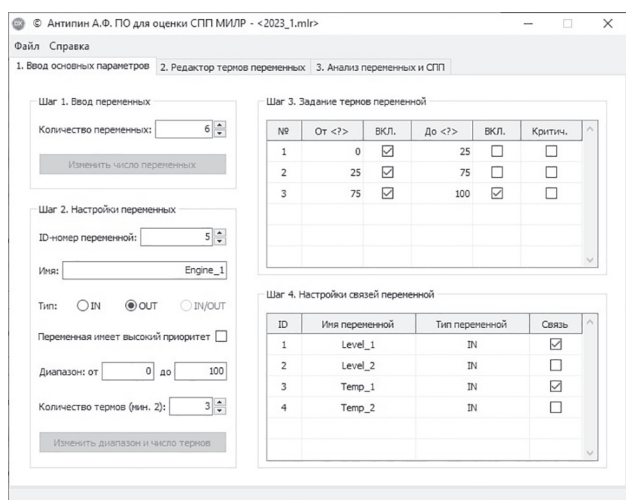


Рис. 5. Редактор термов непрерывных величин многомерных интервально-логических регуляторов и их взаимосвязей (а) и окно с результатами анализа переменных и системы производственных правил многомерных интервально-логических регуляторов (б)

Fig. 5. Editor of terms of continuous quantities of multidimensional interval-logical regulators and their interrelations (a) and a window with the results of the analysis of variables and the system of production rules of multidimensional interval-logical regulators (b)

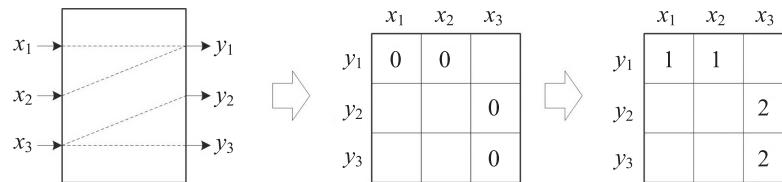


Рис. 6. Схема определения групп взаимосвязанных переменных
 Fig. 6. Scheme of determining groups of interrelated variables

жащие переменные, где отдельные термы отмечены флагом «критический».

Общее количество групп G определено автоматически путем формирования матрицы взаимосвязей по следующей схеме [17]:

- создана матрица размерностью $n \times (m + k)$, где n, m, k — количество входных и выходных непрерывных величин (переменных) МИЛР x, y, z соответственно;
- ячейки на пересечении взаимосвязанных переменных отмечены символом «0», что демонстрирует их принадлежность к некоторой группе.

На рис. 6 показан механизм определения групп взаимосвязанных переменных МИЛР с тремя входными (x_1, x_2, x_3) и тремя выходными (y_1, y_2, y_3) величинами, где символами «1» и «2» отмечены номера групп.

Содержимое ячеек последовательно обработано по алгоритму.

Шаг 1. Если ячейка содержит символ «0», то в нее записывается номер очередной группы, начиная с единицы. В противном случае выполняется переход к шагу 2.

Шаг 2. Выполняется проверка значений, записанных в ячейках, находящихся с текущей обрабатываемой ячейкой на одной линии по вертикали и по горизонтали, т. е. входящих в один и тот же столбец или строку.

Если в какой-либо ячейке содержится символ «0», то его заменяют на номер группы.

Шаг 3. Осуществляется переход к шагу 1 до тех пор, пока все ячейки матрицы не будут обработаны.

3) **Количество критических правил для каждой группы $R_{гр.кр}^g$.** Данный параметр позволяет выделить в СПП особо важные правила, в которых фигурируют термы, отмеченные как критические, и может быть рассчитан следующим образом:

$$R_{гр.кр}^g = R_{гр}^g - \prod_{i=1}^{n_g} (K(x_i) - K_{кр}(x_i)) \prod_{j=1}^{n_g} (K(z_j) - K_{кр}(z_j)),$$

где $K_{кр}$ — количество критических термов, определенных для входных непрерывных величин x и z МИЛР в количестве n_g , образующих группу с порядковым номером g ; $R_{гр}^g$ — максимальное число продукционных правил группы:

$$R_{гр}^g = \prod_{i=1}^{n_g} K(x_i) \prod_{j=1}^{n_g} K(z_j).$$

На рис. 7 представлен вид приложения для тестирования и симуляции работы разработанной программы и симуляции работы МИЛР. Симулятор выполняет автоматический расчет значений выходных величин МИЛР при конкретных значениях входных, а также строит графики, отобража-

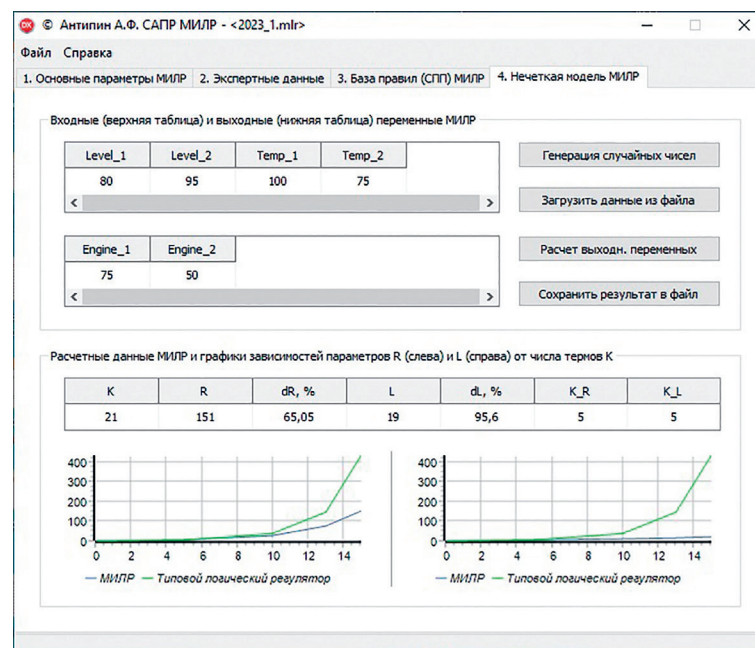


Рис. 7. Приложение для тестирования и симуляции работы многомерных интервально-логических регуляторов
 Fig. 7. Application for testing and simulating the operations of multidimensional interval-logical regulators

ющие изменение отдельных параметров (по сравнению с логическими регуляторами, блок логического вывода которых работает по схеме, подразумевающей обработку всех продукционных правил) [15–17].

Симулятор позволяет также внести коррективы в алгоритмы деинтервализации и менять приоритеты отдельных переменных и их термов.

Также возможна автоматическая генерация продукционных правил на основе имеющихся наборов экспертных данных в виде совокупности значений входных и выходных непрерывных величин МИЛР в отдельные моменты времени t . Эти данные используются в качестве контрольных точек, определяющих центр термов. Диапазоны значений термов подбираются автоматически, исходя из соображений равномерного распределения между контрольными точками.

Описанные редакторы и приложения формируют единую среду для быстрой разработки программ МИЛР с общей концепцией и форматом файлов, что позволяет использовать их в тесной связи друг с другом, а также в качестве самостоятельных программных единиц.

Заключение

Рассмотрен новый подход к разработке многомерных нечетких регуляторов в специальной визуальной среде. Среда обеспечивает наиболее наглядное и быстрое создание функциональных блоков регуляторов для программируемых контроллеров. Получена возможность анализа и симуляции работы системы продукционных правил и регуляторов в целом, что позволяет существенно сократить сроки разработки и отладки систем управления на их основе.

Разработанное специальное программное обеспечение создано и протестировано в интегрированной среде разработки Embarcadero Delphi для операционной системы Microsoft Windows версии 7 и старше. Программное обеспечение может использоваться в работе специалистами, связанными с разработкой систем управления сложными технологическими процессами и объектами при отсутствии адекватной математической модели.

Литература

1. Филимонов А.Б., Филимонов Н.Б. Некоторые проблемные аспекты нечеткого ПИД регулирования // Мехатроника, автоматизация, управление. 2018. Т. 19. № 12. С. 762–769. <https://doi.org/10.17587/mau/19.762-769>
2. Bykovtsev Y.A., Lokhin V.M. Estimation of the accuracy of a control system with a fuzzy pid controller based on the approximation of the static characteristic of the controller // Мехатроника, автоматизация, управление. 2021. Т. 22. № 12. С. 619–624. <https://doi.org/10.17587/mau.22.619-624>
3. Kocian J., Koziorek J., Pokorný M. Implementation of fuzzy logic control based on PLC // Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2011. P. 1–8. <https://doi.org/10.1109/ETFA.2011.6059049>
4. Бурakov М.В., Коновалов А.С., Яковец О.Б. Эволюционный синтез нечетких регуляторов // Информационно-управляющие системы. 2015. № 6(79). С. 28–33. <https://doi.org/10.15217/issn1684-8853.2015.6.28>
5. Игнатъев В.В., Соловьев В.В. Метод синтеза нечетких регуляторов на основе кластеризации // Программные продукты и системы. 2021. Т. 34. № 4. С. 597–607. <https://doi.org/10.15827/0236-235X.136.597-607>
6. Богачев И.С., Коломыцев В.Г. Методика настройки нечеткого регулятора с нелинейной характеристикой Р-типа средствами Matlab // Современные наукоемкие технологии. 2020. № 11-1. С. 9–14. <https://doi.org/10.17513/snt.38330>
7. Buriboev A., Kang H.K., Ko M.-C., Oh R., Abduvaitov A., Jeon H.S. Application of fuzzy logic for problems of evaluating states of a computing system // Applied Sciences. 2019. V. 9. N 15. P. 3021. <https://doi.org/10.3390/app9153021>
8. Dumitrescu C., Ciotirnae P., Vizitiu C. Fuzzy logic for intelligent control system using soft computing applications // Sensors. 2021. V. 21. N 8. P. 2617. <https://doi.org/10.3390/s21082617>
9. Усков А.А., Лаврушин В.М. Системы управления с нечеткими супервизорными ПИД-регуляторами // Информационные технологии моделирования и управления. 2015. Т. 93. № 3. С. 279–285.
10. Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления. Теория и практика: учебное пособие. М.: Радиотехника, 2009. 388 с.
11. Пегат А. Нечеткое моделирование и управление. М.: БИНОМ. Лаборатория знаний, 2009. 798 с.
12. Bobyr M.V., Emelyanov S.G. A nonlinear method of learning neuro-fuzzy models for dynamic control systems // Applied Soft Computing. 2020. V. 8. P. 106030. <https://doi.org/10.1016/j.asoc.2019.106030>

References

1. Filimonov A.B., Filimonov N.B. Certain problematic aspects of fuzzy pid regulation. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2018, vol. 19, no. 12, pp. 762–769. (in Russian). <https://doi.org/10.17587/mau/19.762-769>
2. Bykovtsev Y.A., Lokhin V.M. Estimation of the accuracy of a control system with a fuzzy pid controller based on the approximation of the static characteristic of the controller. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2021, vol. 22, no. 12, pp. 619–624. <https://doi.org/10.17587/mau.22.619-624>
3. Kocian J., Koziorek J., Pokorný M. Implementation of fuzzy logic control based on PLC. *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2011, pp. 1–8. <https://doi.org/10.1109/ETFA.2011.6059049>
4. Burakov M.V., Konovalov A.S., Yakovets O.B. Evolutionary design of fuzzy controllers. *Information and Control Systems*, 2015, no. 6(79), pp. 28–33. (in Russian). <https://doi.org/10.15217/issn1684-8853.2015.6.28>
5. Ignatyev V.V., Solovov V.V. A synthesis method for fuzzy controllers based on clustering. *Software & Systems*, 2021, vol. 34, no. 4, pp. 597–607. (in Russian). <https://doi.org/10.15827/0236-235X.136.597-607>
6. Bogachev I.S., Kolomytsev V.G. Technique for adjusting a fuzzy regulator with a nonlinear P-type characteristic by Matlab. *Modern High Technologies*, 2020, no. 11-1, pp. 9–14. (in Russian). <https://doi.org/10.17513/snt.38330>
7. Buriboev A., Kang H.K., Ko M.-C., Oh R., Abduvaitov A., Jeon H.S. Application of fuzzy logic for problems of evaluating states of a computing system. *Applied Sciences*, 2019, vol. 9, no. 15, pp. 3021. <https://doi.org/10.3390/app9153021>
8. Dumitrescu C., Ciotirnae P., Vizitiu C. Fuzzy logic for intelligent control system using soft computing applications. *Sensors*, 2021, vol. 21, no. 8, pp. 2617. <https://doi.org/10.3390/s21082617>
9. Uskov A.A., Lavrushin V.M. Control systems with fuzzy supervisory PID regulators. *Informacionnye tehnologii modelirovaniya i upravleniya*, 2015, vol. 93, no. 3, pp. 279–285. (in Russian)
10. Vasilev V.I., Iliasov B.G. *Intelligent Control Systems. Theory and Practice*. Moscow, Radiotekhnika Publ., 2009, 388 p. (in Russian)
11. Piegat A. *Fuzzy Modeling and Control*. Physica-Verlag HD, 2001, 728 p.
12. Bobyr M.V., Emelyanov S.G. A nonlinear method of learning neuro-fuzzy models for dynamic control systems. *Applied Soft Computing*, 2020, vol. 8, pp. 106030. <https://doi.org/10.1016/j.asoc.2019.106030>
13. Bobyr M.V., Milostnaya N.A., Kulabuhov S.A. A method of defuzzification based on the approach of areas' ratio. *Applied Soft*

13. Bobyр M.V., Milostnaya N.A., Kulabuhov S.A. A method of defuzzification based on the approach of areas' ratio // *Applied Soft Computing*. 2017. V. 59. P. 19–32. <https://doi.org/10.1016/j.asoc.2017.05.040>
14. Karakuzu C., Karakaya F., Çavuşlu M.A. FPGA implementation of neuro-fuzzy system with improved PSO learning // *Neural Networks*. 2016. V. 76. P. 128–140. <https://doi.org/10.1016/j.neunet.2016.02.004>
15. Антипин А.Ф. Особенности программной реализации многомерных логических регуляторов с переменными в виде совокупности аргументов двузначной логики // *Автоматизация и современные технологии*. 2014. № 2. С. 30–36.
16. Антипин А.Ф. Особенности настройки взаимосвязей и составления системы продукционных правил в интервально-логическом регуляторе // *Информационные системы и технологии*. 2015. № 1(87). С. 5–13.
17. Антипин А.Ф. Об оценке программ контроллеров с программируемой логикой // *Автоматизация, телемеханизация и связь в нефтяной промышленности*. 2018. № 11. С. 41–46. <https://doi.org/10.30713/0132-2222-2018-11-41-46>
18. Antipin A.F. Features of the multidimensional logical regulators software implementation with variables as a set of binary logic arguments. *Avtomatizaciya i sovremennye tehnologii*, 2014, no. 2, pp. 30–36. (in Russian)
19. Antipin A.F. Features of the setup of interlinkages and the creation of the system of condition-action rules in interval-logic controller. *Informacionnye sistemy i tehnologii*, 2015, no. 1(87), pp. 5–13. (in Russian)
20. Antipin A.F. Programs evaluation of controllers with programmable logics. *Automation and Informatization of the fuel and energy complex*, 2018, no. 11, pp. 41–46. (in Russian). <https://doi.org/10.30713/0132-2222-2018-11-41-46>

Авторы

Антипин Андрей Федорович — кандидат технических наук, доцент, доцент, Стерлитамакский филиал Уфимского университета науки и технологий, Стерлитамак, 453103, Российская Федерация, [sc 55904921400](https://orcid.org/0000-0002-9151-4167), <https://orcid.org/0000-0002-9151-4167>, andrejantipin@ya.ru

Антипина Евгения Викторовна — кандидат физико-математических наук, старший научный сотрудник, Уфимский университет науки и технологий, Уфа, 450076, Российская Федерация, [sc 57214879770](https://orcid.org/0000-0002-8458-9638), <https://orcid.org/0000-0002-8458-9638>, stepashinaev@ya.ru

Статья поступила в редакцию 01.07.2023
Одобрена после рецензирования 31.07.2023
Принята к печати 21.09.2023

Authors

Andrey F. Antipin — PhD, Associate Professor, Associate Professor, Sterlitamak branch of Ufa University of Science and Technology, Sterlitamak, 453103, Russian Federation, [sc 55904921400](https://orcid.org/0000-0002-9151-4167), <https://orcid.org/0000-0002-9151-4167>, andrejantipin@ya.ru

Evgenia V. Antipina — PhD (Physics&Mathematics), Senior Researcher, Ufa University of Science and Technology, Ufa, 450076, Russian Federation, [sc 57214879770](https://orcid.org/0000-0002-8458-9638), <https://orcid.org/0000-0002-8458-9638>, stepashinaev@ya.ru

Received 01.07.2023
Approved after reviewing 31.07.2023
Accepted 21.09.2023



Работа доступна по лицензии
 Creative Commons
 «Attribution-NonCommercial»