

УДК 004.896

Т.Г. Петренко, О.С. Тимчук

Донецкий национальный университет, Донецк

ПАКЕТ БИБЛИОТЕК И СРЕДА ПОДДЕРЖКИ НЕЧЕТКИХ ДИСКРЕТНЫХ ИНТЕРВАЛЬНЫХ ЛОГИЧЕСКИХ СИСТЕМ ВТОРОГО ТИПА

В работе представлен программный продукт поддержки нечетких дискретных интервальных логических систем второго типа. Предложена декомпозиция нечеткого дискретного интервального логического вывода второго типа с унификацией модулей обработки. Для организации удобного интерфейса входные и выходные параметры модулей представлены в табличном виде, для их хранения и обработки предложен XML-формат. Программный продукт реализует все фазы нечеткого логического вывода и может быть интегрирован в приложения, которые реализуют нечеткую систему принятия решений.

Ключевые слова: нечеткое дискретное интервальное множество второго типа, нечеткая дискретная интервальная система второго типа, инструментарий поддержки нечетких множеств и систем второго типа, пакет библиотек.

Введение

Увеличение объемов публикаций по дискретным интервальным нечетким множествам второго типа (DIT2FSs) говорит о реальном интересе к описанию сложных систем с помощью аппарата DIT2FS [1]. Однако существующие инструментальные средства работы с DIT2FS ограничены либо коммерческими программными продуктами, либо решением узких задач, либо расширением существующих возможностей пакета MatLab, что затрудняет дальнейшее использование модели на базе DIT2FS [2 – 5]. Актуальным является создание универсальных инструментов (библиотек) поддержки аппарата DIT2FS в нечетких системах принятия решений.

В работе предложена декомпозиция дискретного интервального нечеткого логического вывода второго типа (DIT2FLS), реализованы пакет библиотек и среда поддержки DIT2FLS [6]. Пакет библиотек имеет прозрачную структуру, содержит универсальные структуры данных, является расширяемым и полным. Перечисленные качества пакета позволяют легко подключать его к приложениям, расширять функциональность библиотек, не изменяя архитектурной целостности пакета.

Постановка задачи. Разработать пакет библиотек (DIT2FLS Package Library) и среду поддержки (DIT2FLS Toolbox) DIT2FLSs. Для анализа, про-

ектирования и моделирования систем на основе нечеткой логики второго типа обеспечить необходимой функциональностью DIT2FLS Toolbox. DIT2FLS Package Library должен поддерживать все фазы нечеткого логического вывода [7] и иметь прозрачную, расширяемую архитектуру, что позволит подключать его в другие приложения. При интегрировании DIT2FLS Package Library в приложения, пользователю должен предоставляться удобный формат представления входных и выходных параметров модели.

Решение задачи

DIT2FLS состоит из шести блоков (рис. 1): операция фаззификации (Fuzzifier), блок лингвистических переменных (Linguistic variables), база правил (Rules), нечеткий логический вывод (Inference), операция понижения типа (Type-Reducer), операция дефаззификации (Defuzzifier). Представим модель DIT2FLS на основе декомпозиции данных в табличном виде. На вход DIT2FLS подается набор четких числовых значений IN , состоящий из n элементов.

$$IN = \langle in_1, \dots, in_n \rangle, \quad i = \overline{1, n}. \quad (1)$$

Определение термов входных лингвистических переменных и термов результирующей лингвистической переменной выполняется в блоке Linguistic variables (рис. 1).

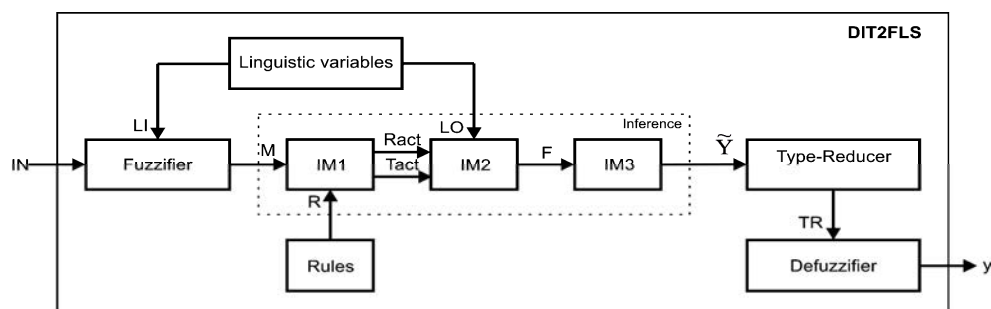


Рис. 1. Структура DIT2FLS

Каждая входная лингвистическая переменная LI_i представляется набором

$$LI_i = \left\langle \tilde{A}_1^{(i)}, \dots, \tilde{A}_j^{(i)}, \dots, \tilde{A}_{d_i}^{(i)} \right\rangle, j = \overline{1, d_i}, i = \overline{1, n}, \quad (2)$$

где d_i – количество терм-множеств i -й лингвистической переменной, $\tilde{A}_j^{(i)}$ – j -е терм-множество i -й лингвистической переменной.

LI_i задана на универсальном множестве X_i с частотой дискретизации h_i . Каждое терм-множество $\tilde{A}_j^{(i)}$ из (2) является DIT2FS. Представим $\tilde{A}_j^{(i)}$ в табличной форме (табл. 1).

Таблица 1
Табличное представление DIT2FS входных лингвистических переменных

| Номер разбиения | $\underline{\mu}(x_k^{(i)})$ | $\overline{\mu}(x_k^{(i)})$ |
|-----------------|----------------------------------|---------------------------------|
| 1 | $\underline{\mu}(x_1^{(i)})$ | $\overline{\mu}(x_1^{(i)})$ |
| ... | | |
| k | $\underline{\mu}(x_k^{(i)})$ | $\overline{\mu}(x_k^{(i)})$ |
| ... | | |
| N_i | $\underline{\mu}(x_{N_i}^{(i)})$ | $\overline{\mu}(x_{N_i}^{(i)})$ |

где $N_i = X_i / h_i$,

$$x_k^{(i)} \in X_i, x_1^{(i)} \leq \dots \leq x_k^{(i)} \leq \dots \leq x_{N_i}^{(i)}, k = \overline{1, N_i},$$

$$\underline{\mu}(x_k^{(i)}) \leq \overline{\mu}(x_k^{(i)}).$$

Результирующая переменная LO представляет набором

$$LO = \left\langle \tilde{F}_1, \dots, \tilde{F}_t, \dots, \tilde{F}_{d_{res}} \right\rangle, t = \overline{1, d_{res}}, \quad (3)$$

где d_{res} – количество терм-множеств результирующей лингвистической переменной LO, \tilde{F}_t – t -е терм-множество лингвистической переменной LO.

LO задана на универсальном множестве X_{res} с частотой дискретизации h_{res} . Каждое терм-множество \tilde{F}_t из (3) является DIT2FS и поэтому может быть представлено аналогично описанию DIT2FS входных лингвистических переменных (табл. 1).

Нечеткая база знаний содержит b правил следующей формы

$$R_p : \text{IF } in_1 \text{ is } \tilde{A}_{j,j \in [1, d_1]}^{(1)} \text{ AND } \dots \text{ AND } \\ in_i \text{ is } \tilde{A}_{j,j \in [1, d_i]}^{(i)} \text{ AND } \dots \text{ AND } \\ in_n \text{ is } \tilde{A}_{j,j \in [1, d_n]}^{(n)} \\ \text{THEN } y_p = \tilde{F}_{t,t \in [1, d_{res}]}; p = \overline{1, b}.$$

Представим базу правил в табличной форме. В табл. 2, колонки 1..n соответствуют входным лингвистическим переменным, $n+1$ колонка (res) – результирующей лингвистической переменной, $n+2$ колонка – весу правила, термы лингвистических переменных представлены своими индексами: j -й индекс соответствует номеру термина LI_i (2), t -й – номеру термина LO (3). Строка в табл. 2 соответствует отдельному правилу (4).

Таблица 2

Табличное представление базы правил DIT2FLS (R)

| 1 | ... | i | ... | n | res | Вес правила |
|---------------------|-----|---------------------|-----|---------------------|-------------------------|-------------|
| $j, j \in [1, d_1]$ | ... | $j, j \in [1, d_i]$ | ... | $j, j \in [1, d_n]$ | $t, t \in [1, d_{res}]$ | v_1 |
| ... | ... | ... | ... | ... | ... | ... |
| $j, j \in [1, d_1]$ | ... | $j, j \in [1, d_i]$ | ... | $j, j \in [1, d_n]$ | $t, t \in [1, d_{res}]$ | v_p |
| ... | ... | ... | ... | ... | ... | ... |
| $j, j \in [1, d_1]$ | ... | $j, j \in [1, d_i]$ | ... | $j, j \in [1, d_n]$ | $t, t \in [1, d_{res}]$ | v_b |

Рассмотрим блок Fuzzifier (рис. 1), который вычисляет степени принадлежности числовых значений входных параметров IN (1) входным нечетким терм-множествам $\tilde{A}_{j,j=1,d_i}^{(i,i=\overline{1,n})}$ (табл. 1). На выходе блока Fuzzifier получаем таблицу степеней принадлежности M (табл. 3), которая формируется из множества всех наборов $\left\langle i, j, \underline{\mu}(x_k^{(i)}), \overline{\mu}(x_k^{(i)}) \right\rangle$, которые удовлетворяют условию

$$\left(\left(in_i = x_k^{(i)} \text{ AND } \overline{\mu}(x_k^{(i)}) > 0 \right) \mid k \in [1, N_i] \right), i = \overline{1, n}. \quad (5)$$

Таблица 3
Таблица степеней принадлежности числовых значений in_i входным нечетким терм-множествам $\tilde{A}_j^{(i)}$ (M)

| Номер LI | Номер термина в LI | $\underline{\mu}(x_k^{(i)})$ | $\overline{\mu}(x_k^{(i)})$ |
|-------------------|---------------------|------------------------------|-----------------------------|
| $i, i \in [1, n]$ | $j, j \in [1, d_i]$ | $\tilde{A}_j^{(i)}[k, 2]$ | $\tilde{A}_j^{(i)}[k, 3]$ |
| ... | ... | ... | ... |
| $i, i \in [1, n]$ | $j, j \in [1, d_i]$ | $\tilde{A}_j^{(i)}[k, 2]$ | $\tilde{A}_j^{(i)}[k, 3]$ |
| ... | ... | ... | ... |
| $i, i \in [1, n]$ | $j, j \in [1, d_i]$ | $\tilde{A}_j^{(i)}[k, 2]$ | $\tilde{A}_j^{(i)}[k, 3]$ |

Механизм нечеткого логического вывода (рис. 1) состоит из следующих блоков: IM1 – блок, вычисляющий степень выполнения каждого правила R_p в отдельности; IM2 – блок, вычисляющий активизированные функции принадлежности заключений каждого правила $Ract$; IM3 – блок, вычисляющий результирующую функцию принадлежности \tilde{Y} выходного значения на основе активизированных заключений отдельных правил.

На выходе блока IM1 получаем таблицу активизированных правил $Ract$ (структура таблицы $Ract$ аналогична табл. 2) и таблицу степени выполнения активизированных правил $Tact$ (табл. 4). Рассмотрим алгоритм формирования таблиц $Ract$ и $Tact$.

Если для правила R_p найдется такой набор элементов $M_1, \dots, M_i, \dots, M_n$, где каждый элемент является строкой из таблицы M (табл. 3), такой что

$$M_1 \neq \dots \neq M_i \neq \dots \neq M_n, M_i \in M, i = \overline{1, n} \quad (6)$$

и

$$\begin{aligned} & (R_p[1] = M_1[2] \text{ AND } M_1[1] = 1) \text{ AND } \dots \text{ AND} \\ & (R_p[i] = M_i[2] \text{ AND } M_i[1] = i) \text{ AND } \dots \text{ AND} \quad (7) \\ & (R_p[n] = M_n[2] \text{ AND } M_n[1] = n), \end{aligned}$$

то в таблицу $Ract$ добавляем строку R_p , а в таблицу $Tact$ добавляем набор $M_1, \dots, M_i, \dots, M_n$.

Таблица 4

Таблица степени выполнения активизированных правил ($Tact$)

| 1 | ... | i | ... | n |
|------------------|-----|------------------|-----|------------------|
| $M_1, M_1 \in M$ | ... | $M_i, M_i \in M$ | ... | $M_n, M_n \in M$ |
| ... | ... | ... | ... | ... |
| $M_1, M_1 \in M$ | ... | $M_i, M_i \in M$ | ... | $M_n, M_n \in M$ |
| ... | ... | ... | ... | ... |
| $M_1, M_1 \in M$ | ... | $M_i, M_i \in M$ | ... | $M_n, M_n \in M$ |

На выходе блока IM2 получаем таблицу F активизированных функций принадлежности заключений каждого правила $Ract$. Рассмотрим алгоритм формирования таблицы F .

Каждый элемент таблицы $Tact$ является набором $\langle i, j, \underline{\mu}(x_k^{(i)}), \overline{\mu}(x_k^{(i)}) \rangle$, то есть

$$Tact_{s,i} = \langle i, j, \underline{\mu}(x_k^{(i)}), \overline{\mu}(x_k^{(i)}) \rangle, s = \overline{1, h},$$

где h – количество строк в таблице $Tact$.

Поэтому для каждого активизированного правила $Ract_s$ определяем:

$$\min \underline{\mu}_k = \underset{i=1, n}{\text{MIN}} (Tact_{s,i} [3]), \quad (8)$$

$$\min \overline{\mu}_k = \underset{i=1, n}{\text{MIN}} (Tact_{s,i} [4]) \quad (9)$$

Согласно (3), (4) и табл. 2:

$$\tilde{F}_s^* = \tilde{F}_i, t = Ract_s [n+1], \quad (10)$$

$$(\tilde{F}_s^*)_k = (\tilde{F}_s^* [k, 1]),$$

$$\text{MIN}(\tilde{F}_s^* [k, 2], \min \underline{\mu}_k) * Ract_s [n+2], \quad (11)$$

$$\text{MIN}(\tilde{F}_s^* [k, 3], \min \overline{\mu}_k) * Ract_s [n+2], k = \overline{1, N_{res}}.$$

Добавляем каждое полученное нечеткое множество \tilde{F}_s^* в таблицу F .

На выходе блока IM3 получаем результирующую функцию принадлежности \tilde{Y} выходного значения на основе активизированных заключений отдельных правил. \tilde{Y} является DIT2FS и определено на универсальном множестве X_{res} , поэтому может быть определено аналогично терм - множествам LO (табл. 1). Рассмотрим алгоритм формирования \tilde{Y} .

$$\tilde{Y}[k, 1] = k, k = \overline{1, N_{res}},$$

$$\tilde{Y}[k, 2] = \underset{s=1, h}{\text{MAX}} (\tilde{F}_s^* [k, 2]), k = \overline{1, N_{res}}, \quad (12)$$

$$\tilde{Y}[k, 3] = \underset{s=1, h}{\text{MAX}} (\tilde{F}_s^* [k, 3]), k = \overline{1, N_{res}}.$$

Рассмотрим блок Type-Reducer (рис. 1), который выполняет понижения типа DIT2FS. Понижение типа реализовано с помощью расширенного итерационного алгоритма Karnik – Mendel (ЕКМ) для интервальных нечетких множеств и систем второго типа [8]. Перед использованием ЕКМ необходимо выполнить следующие подготовительные работы:

1) Удалить из \tilde{Y} все строки, которые удовлетворяют условию $\underline{\mu}(x_k^{(res)}) = 0, x_k^{(res)} \in X_{res}$.

2) Если $N_{res} = 0$, то прекратить вычисления.

3) Если $N_{res} = 1$, то $L = R = x_1^{(res)}, x_1^{(res)} \in X_{res}$.

4) Если $\underset{k=1, N_{res}}{\text{MAX}} (\underline{\mu}(x_k^{(res)})) = 0$, то \tilde{Y} следует

рассматривать как нечеткое множество первого типа, функция принадлежности которого совпадает с верхней границей \tilde{Y} и операцию понижение типа не выполнять.

В результате итерационного алгоритма ЕКМ получаем левую (L) и правую (R) конечные точки.

Блок Defuzzifier (рис. 1) вычисляет четкое выходное значение y по формуле (по умолчанию):

$$y = \frac{x_L^{(res)} + x_R^{(res)}}{2},$$

где $x_L^{(res)} \in X_{res}, x_R^{(res)} \in X_{res}$.

Предложенная декомпозиция DIT2FLS с унификацией блоков обработки, а также структуры входных и выходных данных позволила реализовать DIT2FLS Package Library и DIT2FLS Toolbox с гибкой настройкой процесса нечеткого вывода второго типа.

Программная реализация

DIT2FLS Package Library состоит из 9 модулей (табл. 5). Реализация модулей DIT2FLS Package Library в виде динамических библиотек позволяет заменять их на пользовательские при условии соблюдения интерфейса библиотеки.

DIT2FLS Toolbox состоит из пяти модулей: редактор лингвистических переменных, редактор функций принадлежности лингвистических переменных, редактор базы правил, симулятор, редактор настроек модели. В редакторе лингвистических переменных (рис. 2, а) пользователь определяет входные

и результирующую лингвистические переменные (имя, область определения, частота дискретизации).

Редактор функций принадлежности позволяет задать информацию о термах лингвистической переменной (рис. 2, б) (количество, наименование, тип, параметры). Доступны стандартные функции принадлежности: двухсторонняя гауссовская (gauss2mf), гауссовская (gaussmf), π -подобная (pimf), s-подобная (smf), трапециевидная функция (trapmf), треугольная (trimf), z-подобная (zmf). Редактор также позволяет задавать пользовательские функции принадлежности в дискретном виде.

Таблица 5

Описание модулей пакета библиотек DIT2FLS Toolbox

| Название модуля | Назначение модуля | Входные параметры модуля | Выходные параметры модуля |
|-----------------|---|--------------------------|---------------------------|
| LI.dll | чтение/запись входных лингвистических переменных из/в файл | файл описания LI (XML) | таблица LI |
| LO.dll | чтение/запись результирующей лингвистической переменной из/в файл | файл описания LO (XML) | таблица LO |
| Rules.dll | чтение/запись базы правил из/в файл | файл описания R (XML) | таблица R |
| Fuzzifier.dll | реализация операции фаззификации | таблица LI, набор IN | таблица M |
| IM1.dll | вычисление степени выполнения каждого правила базы правил в отдельности | таблицы M, R | таблицы Ract, Tact |
| IM2.dll | вычисление активизированных функций принадлежности | таблицы Ract, Tact, LO | таблица F |
| IM3.dll | вычисление результирующей функции принадлежности | таблица F | таблица \tilde{Y} |
| EKM.dll | реализация операции понижения типа | таблица \tilde{Y} | результат EKM (L, R) |
| Defuzzifier.dll | реализация операции дефаззификации | результат EKM (L, R) | результат y |



Рис. 2. Интерфейс DIT2FLS Toolbox: а – редактор лингвистических переменных; б – редактор функций принадлежности лингвистических переменных; в – редактор базы правил

Редактор базы правил предназначен для формирования и модификации нечетких правил (рис. 2, в). Редактор содержит поддержку логики И, ИЛИ и НЕ в настраиваемых правилах. База правил может задаваться как пользователем, так и генерироваться автоматически последовательным перебором термов лингвистических переменных.

Симулятор позволяет выполнять интерактивный просмотр нечеткого логического вывода.

Редактор настроек модели позволяет изменять настройки работы DIT2FLS Toolbox.

DIT2FLS Toolbox поддерживает два алгоритма логического вывода – Mamdani (по умолчанию) и

Sugeno. Операция понижения типа реализована с помощью алгоритма EKM. Методы дефаззификации – центр тяжести (по умолчанию), центр максимумов, первый максимум.

DIT2FLS Package Library и DIT2FLS Toolbox (программный комплекс) не накладывают ограничения на количество лингвистических переменных, термов лингвистических переменных и размер базы правил (ограничение только по допустимому объему памяти компьютера).

Программный комплекс позволяет выполнять чтение/запись входных параметров модели (лингвистические переменные, база правил) в/из файл (а)

в XML форматі. Представимо XML - структуру описання входних лінгвістических перемінних моделі DIT2FLS (рис. 3).

```
<DIT2FLS>
<systemInfo>
  <NumLIs>Кол-во LI</NumLIs>
</systemInfo>
<LI id="імя перемінної">
  <x>значення  $x_k$ , роздільник ";"</x>
  <NumFSs>кількість термів</NumFSs>
  <MF id="імя терма">
    <lmf>значення  $\underline{\mu}(x_k^{(i)})$ , роздільник ";"</lmf>
    <umf>значення  $\bar{\mu}(x_k^{(i)})$ , роздільник ";"</umf>
  </MF>
  ...
</LI>
...
</DIT2FLS>
```

Рис. 3. XML-структура описання входних лінгвістических перемінних моделі

Для розробки програмного комплексу використовувалися Microsoft Visual Studio 2010, технологія .NET, мовою програмування С#.

Требования для експлуатації програмного комплексу:

1. Операційна система: починаючи з Windows XP з пакетом оновлення 3 (SP3) (тільки 32-розрядні).
2. Microsoft .NET Framework 4.0

Висновки

1. Виконано декомпозиція DIT2FLS з уніфікацією блоків обробки та декомпозиція структур входних та вихідних даних DIT2FLS. Для представлення входних параметрів DIT2FLS розроблено XML-формат.

2. Розроблено пакет бібліотек та середовище підтримки DIT2FLSs. Програмний комплекс може

бути використаний у різних прикладних областях. Архітектура пакета бібліотек дозволяє підключати його в програми, модулі пакета можуть замінюватися користувальницькими. При інтегруванні пакета бібліотек або модулів пакета необхідно дотримуватися формату входних та вихідних параметрів, вимоги до експлуатації та інтерфейсу бібліотек.

Список літератури

1. Mendel Jerry M. Type-2 fuzzy sets made simple / Jerry M. Mendel, R.I.B. John // IEEE Trans. Fuzzy Syst. – 2002. – Vol. 10, No. 2. – P. 117-127.
2. Toolbox for Interval Type-2 Fuzzy Logic Systems / [Zamani M., Nejati H., Jahromi A.T. and other] // Proceedings of the 11th Joint Conference on Information Sciences JCIS. – 2008. – P. 1- 6.
3. Castillo O. Computational intelligence software: Type-2 Fuzzy Logic and Modular Neural Networks / O. Castillo, P. Melin // Neural Networks (IJCNN), The 2011 International Joint Conference. – 2008. – P. 1820-1827.
4. Type-2 Fuzzy Logic Software / [Mendel Jerry M., Karnik Nilesh N., Liang Q. and other] [Електронний ресурс]. Режим доступу до ресурсу: <http://sipi.usc.edu/~mendel/software/> — Дата доступу: 8.11.2011.
5. Aria M. INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS TOOLBOX / M. Aria // Majalah Ilmiah UNIKOM. – 2011. – Vol. 9, No1. – P. 43-49.
6. Пакет бібліотек та середовище підтримки дискретних інтервальних нечітких логічних систем другого типу [Електронний ресурс]. Режим доступу до ресурсу: <https://sites.google.com/site/dit2fls/> — Дата доступу: 8.11.2011.
7. Пегат А. Нечітке моделювання та управління / А. Пегат. – М.: БИНОМ. Лабораторія знань, 2009. – 798 с.
8. Wu D. Enhanced Karnik-Mendel Algorithms for Interval Type-2 Fuzzy Sets and Systems / D. Wu, Jerry M. Mendel // Fuzzy Information Processing Society. – 2007. – P. 184-189.

Поступила в редакцію 18.11.2011

Рецензент: д-р техн. наук, проф. А.А. Каргин, Донецький національний університет, Донецьк.

ПАКЕТ БІБЛІОТЕК І СЕРЕДОВИЩЕ ПІДТРИМКИ НЕЧІТКИХ ДИСКРЕТНИХ ІНТЕРВАЛЬНИХ ЛОГІЧНИХ СИСТЕМ ДРУГОГО ТИПУ

Т.Г. Петренко, О.С. Тимчук

У роботі представлений програмний продукт підтримки нечітких дискретних інтервальних логічних систем другого типу. Запропоновано декомпозиція нечіткого дискретного інтервального логічного виводу другого типу з уніфікацією модулів обробки. Для організації зручного інтерфейсу вхідні та вихідні параметри модулів представлені в табличному вигляді, для їх зберігання та обробки запропонований XML-формат. Програмний продукт реалізує всі фази нечіткого логічного виводу і може бути інтегрований в додатки, які реалізують нечітку систему прийняття рішень.

Ключові слова: нечітка дискретна інтервальна множина другого типу, нечітка дискретна інтервальна система другого типу, інструментарій підтримки нечітких множин та систем другого типу, пакет бібліотек.

LIBRARY PACKAGE AND TOOLBOX FOR DISCRETE INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

T.G. Petrenko, O.S. Timchuk

This paper presents the software supporting discrete interval type-2 fuzzy logic systems. A decomposition of discrete interval type-2 fuzzy inference with the unification of the processing units is proposed in the present paper. Input and output parameters of the units are presented in tabular form for convenient organization of programming interface. And XML – format is proposed for holding and processing this parameters. The software provides all stages of fuzzy inference and can be integrated into an application which implements fuzzy system of making decision.

Keywords: discrete interval type-2 fuzzy set, discrete interval type-2 fuzzy system, software for discrete interval type-2 fuzzy sets and systems, library package.