УДК 519.853

Zelentsov D.G., Naumenko N.Yu., Gorbonos P.A.

SOFTWARE FOR DESIGN SYSTEMS OF FUZZY CONTROL AND CONDUCTING FUZZY-MULTIPLE CALCULATIONS

Ukrainian State University of Chemical Technology, Dnipro

The paper suggests a description of the design concept and ways to implement the library to work with data that are fuzzy. Based on a review of the latest researches in the field of fuzzy calculations, we formulated the basic requirements for software for performing fuzzy-multiple calculations. The aim of the work is the development of a software implemented as a modular library that satisfies the requirements of universality, integrability, speed, portability, completeness and independence. To ensure the flexibility of the library, the fuzzy data are differentiated into fuzzy sets and fuzzy numbers. We presented the implementation of a multi-purpose library for working with fuzzy data. We offered the description of the main stages of software design, the tool for their creation and the structure of library modules. We made a class diagram that shows the relationships in the library, and a description of the function modules for working with these classes. The developed software can be integrated into an existing or projected control system.

Keywords: fuzzy set, fuzzy control system, library.

Statement of the problem

In 1965, the Azerbaijani mathematician Lotfi Zadeh published the fundamental work on the theory of fuzzy sets, having based a separate direction of mathematics. In 1973, he proposed a theory of fuzzy logic, which initiated the application of the new mathematical theory in practice. Since then, the new mathematical apparatus has firmly entered the engineering and applied practice in many branches of science and technology. As examples of the use of fuzzy logic in practice, one can give the automatic gate control system of a dam in a hydroelectric power plant (Tokyo Electric Power), the aiming of cameras during the broadcast of sports events (Hirota, Fuji, Toshiba, Omron), control of air conditioning systems (Mitsubishi, Sharp), engine control and economical vehicle speed (Nissan), semiconductor manufacturing control (Canon), document archiving (Mitsubishi), character recognition (Sony), motion recognition (Canon) and many others [1]. The scope of fuzzy control systems, elements of fuzzy logic or at least its principles is extremely wide and continues to expand. In this connection, the problem arises of the tools for designing and developing fuzzy control systems. Today a large number of individual programs and complexes for fuzzy calculations have been produced. But existing software products and complexes for the development of fuzzy systems are not flexible enough to include fuzzy calculations in a separate project, in the overwhelming majority they are commercial products, mostly focused on solving specific problems. Often, to build a project on the principles of fuzzy logic, it is necessary to develop from scratch the calculated part for the current task, which causes problems with the repeatability of using the resulting module. Therefore, the problem of the existence of a flexible open universal mechanism for developing fuzzy systems remains open.

Analysis of recent research and publications

During the existence of fuzzy calculations, a fairly large number of instruments were developed to implement fuzzy control systems.

One of the first to mention is the Fuzzy Logic Toolbox, which is part of MatLab environment. This software allows you to build fuzzy approximating systems, fuzzy expert systems, hybrid (neural network) systems, gives the opportunity to build quite complex rules systems, has a wide range of membership functions. At the same time, this software has a convenient graphical interface, in spite of this, you can work with it from the command line using an extensive set of commands. The software can also be used in the design of systems in Simulink environment [2-4].

One of the best-selling packages for the design of fuzzy systems is CubiCalc software from

HyperLogic. This software can be integrated into Microsoft Excel to perform fuzzy calculations in the form of tables, can be used with programs that support Visual Basic. The company also supplies an additional library for the design of neural networks based on fuzzy sets [5].

FuziCalc of FuziWare is also known as a software for working with fuzzy quantities. This software supports work with membership functions and tables of both exact and fuzzy values [6].

Among the well-known programs are Triumph-Analytic Corporation «Parus», software AnyLogic, ITHINK, PolyAnalyst, ExPro Master and others [7].

All of the above software products, in addition to being corporate and not freely available, have one thing in common: they are separate programs that can only be used to design fuzzy systems and can not be integrated into the finished project. For the project to work, a separate development of the calculation modules is necessary. In addition, all these software tools are not flexible enough to work, for example, with fuzzy arithmetic, since they are oriented to a rule base with fuzzy inference, which makes them at best inconvenient for solving particular problems.

There is a simple expert system FuzzyAdvicer in free access, written in the language F# [8], but it is not universal when working with various fuzzy systems and can not be integrated directly into the developed control system.

A huge disadvantage is also the incompleteness of the provided instruments for calculation. In the overwhelming majority, the possibility of fuzzy inference is given only by Mamdani algorithm, less often – by Sugeno algorithm. Of the existing fuzzy operators, only those based on the minimax t-norm are used. This significantly limits the possibility of calculation.

Formulation of the research objective

The aim of the study is to develop a software that will meet the following requirements:

– universality: the software should provide all the possibilities for design fuzzy control systems, and for fuzzy calculations within the projects being developed;

– integrability: the software should provide the ability to conduct calculations outside its environment, as a component of the system being designed, that is, to perform the duties of a calculation module that transfers data directly to the control system in an automatic mode;

– speed: the software should provide a sufficiently high speed of operation in real conditions;

– portability: the software should provide the ability to reuse in various fuzzy systems on different platforms;

– completeness: the software should provide a full set of possible fuzzy calculations;

– independence: the software should not depend on the details of the implementation of a particular platform.

Statement of the main research material

C++ was used as the programming language, the development was carried out in Visual Studio 2013 programming environment under Windows 7. To maintain universality, portability and independence, the software was designed as a modular library. The advantage of this approach is that the library written in C++ integrates well into almost any developed system, it is supported by quite a large number of environments and programming languages, it is well transferred to other operating systems (in particular – on Mac OS and on Unix/Linux). Since the library for working with fuzzy sets was planned to be independent of the platform implementations, the entire calculation part and the data structures for it were original solutions.

The first stage of development was the implementation of the abstract data type «set». As a basis, a data structure of the «red-black tree» type was written (Fig. 1). A red-black tree is a binary search tree with one additional bit of color in each node.

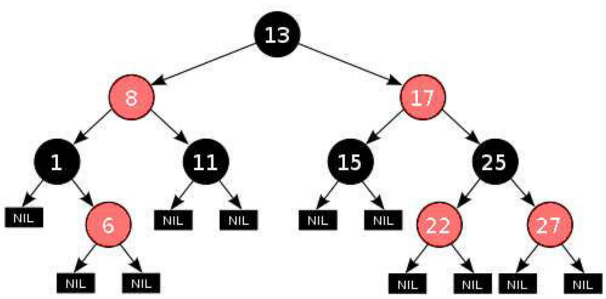

Fig. 1. Example of red-black tree

The color of the node can be either conditional red, or conditional black [9]. A red-black tree must satisfy the following properties:

– each node is either red or black;

– the root of the tree is a black node;

– each leaf of a tree (zero element) is a black node;

– if the node is red, then both children are black;

– for each node, all the simple paths from it to leaves that are descendants of this node contain the same number of black nodes [9].

These restrictions implement the main property of the red-black tree: the path from the root to the farthest leaf is no more than twice as long as the path from the root to the nearest leaf. As a result, the red-black tree is approximately balanced. Because of this, it satisfies the requirement of speed of operation, since the basic operations when working with this data structure (element search, insertion of a new element, and deletion of an element) occupy no more than O (log n) number of operations.

In this case, the red-black tree very well corresponds to the abstract data type «set», because it allows to store data in each node in the form necessary for us and at the same time excludes the appearance of duplicate elements of the set. All the operations necessary for working with the tree (delete, insert, search, search for the next, previous, etc.) were implemented. At the initial stages of the development of the library, a restriction was accepted that the used sets are finite and to each element there corresponds a certain value of the membership function, that is, the membership function is given to the set in the form of a table of values. Therefore, each node of the received tree contains information about the stored element of the set and the value of the membership function. In order to use the resulting class in the future as a container, an additional iterator class was created, based on a red-black tree node.

The resulting container was used in a class that describes the actual data type «set». For this class, the main methods and interfaces were implemented: insertion, deletion, searching for an element by an available key, searching for the smallest/largest, passing the container using an iterator.

Based on the accepted constraint on the finiteness of the set, we developed classes for working with fuzzy elements (Fig. 2). In particular, we carried out the differentiation of fuzzy sets into fuzzy sets themselves and fuzzy numbers, which ensured the flexibility of the library.

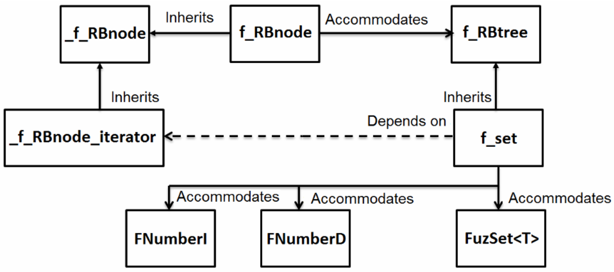

Fig. 2. Diagram of library classes

Fuzzy numbers provide work with interval arithmetic and can be used in private calculations. They are rigidly typed, which greatly facilitates the calculations. The class of the fuzzy set itself is written using template technology, which makes it possible not to tie to the data type, and provides code portability. The library has a modular structure, therefore the user can use only the part necessary to him, even the base class of the data structure «set» is available separately.

Fig. 2 provides a simplified diagram of classes that demonstrates the relationship in the library. The description of classes is given in Table. 1.

Table 1

The description of library classes

Class	Description
_f_RBnode	Class-implementation of a red-black tree node
f_RBnode	Class-interface to the node of red-black tree
f_RBtree	Red-black tree
_f_RBnode_iterator	Iterator for moving on red-black tree
f_set	Set
FNumberI	Fuzzy number on the set of natural numbers
FNumberD	Fuzzy number on the set of real numbers
FuzSet<T>	Class-template of fuzzy set

Table 2

Module structure of the library

Module	Content
Red-black tree	Classes _f_RBnode, f_RBnode, f_RBtree
Set	Module of red-black tree and classes _f_RBnode_iterator, f_set
Standard functions	A set of mathematical functions and text formatting functions to ensure library independence
Fuzzy integer number	Modules of tree, set, standard functions and class FNumberI
Fuzzy real number	Modules of tree, set, standard functions and class FNumberD
Operations on a fuzzy set	A set of functions that provide logical operations (intersection, union, etc.), as well as the choice of various t-norms for these operations
Fuzzy set	Modules of tree, set, standard functions, operations and class FuzSet<T>

```
#pragma once
#include "targetver.h"
#include <stdio.h>
#include <tchar.h>
#include <iostream>

#include "FNumberI.h"
using namespace fuz;

int _tmain(int argc, _TCHAR* argv[])
{
        int mas[3] = { 0, 1, 2 };
//Создаём массив значений элементов нечёткого числа.
        double mmas[3] = { 0.2, 1, 0.2 };
//Создаём массив значений принадлежности элементов нечёткого числа.
        FNumberI num1("Number1", mas, mmas, 3);
//Инициализируем нечёткие числа.
        FNumberI num2("Number2", mas, mmas, 3);
        std::cout << num1.ToString() << "\n";
        num1 += num2;
//Суммируем числа и заносим результат в первое число
        std::cout << num1.ToString() << "\n";
        std::cout << FUnion(&num1, &num2).ToString() << "\n";
//Выводим результат объединения чисел.
        std::cout << FIntersection(&num1, &num2).ToString() << "\n";
//Выводим результат перечесеия чисел.
        std::cout << "First number "<<num1.Defuzzy() << "\tSecond number " << num2.Defuzzy() << "\n";
//Проводим дефаззификацию чисел.
        return 0;
}
```

Fig. 3. An example of a simple program using a library

At the same time, the library in addition to the above classes also contains separate modules of functions for working with these classes. An approximate structure of the modules is given in Table. 2.

Concerning fuzzy logic in the library (Fig. 3), we implemented:

– classes of fuzzy numbers and a fuzzy set itself;

– basic characteristics of fuzzy sets (height, mode, carrier, alpha-level, core, norm);

– the choice of several t-norms for fuzzy operations (min / max, algebraic, boundary, drastic);

– logical operations (union, intersection, complement, equality, difference, symmetric difference, disjunctive sum);

– algebraic operations on fuzzy numbers;

– algebraic actions on fuzzy sets (concentration, extension, exponentiation, multiplication by number, combination);

– operator of increasing fuzziness;

– defuzzification of fuzzy number.

All theoretical information for the realization of the fuzzy set functional is given in [2–4,10].

To fully comply with the requirements set for the library, the following steps are necessary to develop the software:

– add of fuzzy relationships and operations over them;

– add of a linguistic variable, term sets, operations on term sets;

– add of a fuzzy output with a choice of output algorithm;

– add of defuzzification of fuzzy set;

– add of infinite fuzzy sets with a given membership function;

– add of a set of standard membership functions;

– porting the library to Unix/Linux and MacOS systems;

– add of top of the framework library as a separate customizable design tool for fuzzy systems with a graphical interface.

Conclusions

As a result of the work, a library for working with fuzzy sets was designed and implemented. The resulting software tool can be integrated into virtually any fuzzy control system being developed and at the same time makes it possible to perform both small fuzzy calculations and participate in system design.

REFERENCES

1. *Reznik L.* Fuzzy Controllers. – Oxford: Newnes, 1997. – 287 p.

2. *Круглов В.В., Дли М.И., Голунов Р.Ю.* Нечёткая логика и искусственные нейронные сети. – М.: ФИЗМАТЛИТ, 2001. – 201 с.

3. *Леоненков А.А.* Нечёткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. – 736 с.

4. *Штовба С.Д.* Введение в теорию нечётких множеств и нечёткую логику: Монография. – Винница: Континент-Прим, 2003. – 198 с.

5. *HyperLogic* Corporation official site [Электронный ресурс]: [Веб-сайт]. – Электронные данные: Escondido, USA, 2008 – 2016. – Режим доступа: www.hyperlogic.com (дата обращения 08.10.2016). – Название с экрана.

6. *Бочарников В.П.* Fuzzy-технология: Математические основы. Практика моделирования в экономике. – Санкт-Петербург: «Наука» РАН, 2001. – 328 с.

7. *Brainema* [Электронный ресурс]: [Веб-сайт]. – Электронные данные: 20011 – 2016. – Запись от 08.04.2013 – Режим доступа: www.brainema.ru/ 2013/04/fuzzy-logic-programs.html (дата обращения 08.10.2016). – Название с экрана.

8. CyberGuru [Электронный ресурс]: [Веб-сайт]. – Электронные данные: 2006–2014. – Режим доступа: www.cyberguru.ru/algorithms/algorithms-theory/algorithms-fuzzy-logic-system.html (дата обращения 08.10.2016). – Название с экрана.

9. *Алгоритмы:* построение и анализ, 3-е изд. : Пер. с англ. / Томас Кормен, Чарльз Лейзерон, Рональд Ривест, Клиффорд Штайн. – М.: ООО «И. Д. Вильямс», 2013. – 1328 с.

10. *Заде Л.* Нечёткая логика: Понятие лингвистической переменной и его применение к принятию приближённых решений. – М.: Мир, 1976. – 167 с.

Поступила в редакцию 12.10.2016

ПРОГРАМНИЙ ЗАСІБ ДЛЯ ПРОЕКТУВАННЯ СИСТЕМ НЕЧІТКОГО УПРАВЛІННЯ І ПРОВЕДЕННЯ НЕЧІТКО-МНОЖИННИХ РОЗРАХУНКІВ

Зеленцов Д.Г., Науменко Н.Ю., Горбонос П.О.

В роботі пропонується опис концепції проектування і способи реалізації бібліотеки для роботи з даними, що мають нечіткий характер. На підставі огляду останніх досліджень в області нечітких обчислень сформульовані основні вимоги, що пред'являються до програмних засобів для проведення нечітко-множинних розрахунків. Метою роботи є розробка програмного засобу, реалізованого у вигляді модульної бібліотеки, яка задовільняє вимогам універсальності, інтегрованості, швидкодії, переносимості, повноти та незалежності. Для забезпечення гнучкості роботи бібліотеки проведена диференціація нечітких даних на власне нечіткі множини і на нечіткі числа. Представлена реалізація багатоцільової бібліотеки для роботи з нечіткими даними. Пропонується опис основних етапів проектування програмного засобу, інструменту їх створення і структури модулів бібліотеки. Наводиться діаграма класів, що демонструє взаємозв'язки в бібліотеці, і опис модулів функцій для роботи з цими класами. Розроблений програмний засіб може бути інтегрован в уже існуючу або проектовану систему управління.

Ключові слова: нечітка множина, нечітка система управління, бібліотека.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОЕКТИРОВАНИЯ СИСТЕМ НЕЧЁТКОГО УПРАВЛЕНИЯ И ПРОВЕДЕНИЯ НЕЧЁТКО-МНОЖЕСТВЕННЫХ РАСЧЁТОВ

Зеленцов Д.Г., Науменко Н.Ю., Горбонос П.А.

В работе предлагается описание концепции проектирования и способы реализации библиотеки для работы с данными, имеющими нечёткий характер. На основании обзора последних исследований в области нечётких вычислений сформулированы основные требования, предъявляемые к программным средствам для проведения нечётко-множественных расчётов. Целью работы является разработка программного средства, реализованного в виде модульной библиотеки, которая удовлетворяет требованиям универсальности, интегрируемости, быстродействия, переносимости, полноты и независимости. Для обеспечения гибкости работы библиотеки проведена дифференциация нечётких данных на собственно нечёткие множества и на нечёткие числа. Представлена реализация многоцелевой библиотеки для работы с нечёткими данными. Предлагается описание основных этапов проектирования программного средства, инструмента их создания и структуры модулей библиотеки. Приводится диаграмма классов, демонстрирующая взаимосвязи в библиотеке, и описание модулей функций для работы с этими классами. Разработанное программное средство может быть интегрировано в уже существующую или проектируемую систему управления.

Ключевые слова: нечёткое множество, нечёткая система управления, библиотека.