

## ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ОБЪЕКТАМИ И СИСТЕМАМИ

УДК 621.391.Б1

### СИНТЕЗ НЕЧЕТКИХ РЕГУЛЯТОРОВ НА ОСНОВЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЙ

Кондратенко Г. В., Кондратенко Ю. П., Мухортова К. В.

Повышение эффективности управления является актуальной проблемой в условиях возрастающей сложности технологического оборудования, процессов и систем. Для проектирования систем управления сложными объектами важную роль играет решение задач построения адекватных математических или имитационных моделей и синтеза алгоритмов управления, обеспечивающих решение задач в условиях неопределенности. Интеллектуальные системы принятия решений (ИСПР), имитирующие принципы мышления человека, находят все более широкое применение при управлении и формализации сложных объектов и систем. Одним из эффективных подходов, применяемых при синтезе современных ИСПР, является использование теории нечетких множеств и нечеткой логики [1]. Оперирование расплывчатыми понятиями в теории нечетких множеств максимально приближает описание объекта управления к обычной человеческой речи, что значительно облегчает построение модели системы. В свою очередь нечеткое управление реализуется на основе правил в форме «если - то», также интуитивно понятных человеку.

ИСПР на нечеткой логике способны решать множество разноплановых задач управления в условиях неопределенности, в частности при управлении морскими подвижными объектами в экстремальных условиях. Так, например, в практике судовождения это задачи: а) автоматизации процессов принятия решений для своевременного и адекватного реагирования экипажа при управлении судном в экстремальных условиях; б) имитационного моделирования морской среды, нестационарных возмущающих воздействий и соответствующих алгоритмов принятия решений и управления для оптимизации программно-алгоритмического и аппаратного обеспечения судовых ИСПР на этапе их проектирования, в частности на основе вычислительных систем параллельного действия [2].

При создании ИСПР на основе нечеткой логики целесообразно либо разрабатывать программный модуль, реализующий алгоритм нечеткого логического вывода под конкретную задачу, либо использовать готовые программные разработки, предварительно адаптируя их для решения поставленной задачи. Анализ существующего программного обеспечения (ПО) для обработки нечеткой информации позволяет выделить две основные группы ПО такого типа:

- специализированные коммерческие программные продукты (ПП), использующие аппарат нечеткой логики для решения задач в различных отраслях. Такие ПП, к сожалению, не предназначены для кооперации с другими программами в реальном времени (Matlab, Matematica, CubiCalc, RuleMaker, FuziCalc, fuzzyTECH и др.);
- небольшие библиотеки классов с ограниченными возможностями настройки либо разработанные под конкретный программный продукт (Free Fuzzy Logic Library, fuzzyCLIPS).

Перечисленные ПП не обладают достаточным уровнем универсальности, чтобы их компоненты можно было программно встраивать при создании собственных разработок и гибко настраивать в соответствии с требованиями системы.

Данная статья посвящена вопросам создания расширенной объектно-ориентированной библиотеки классов [3] для обработки нечеткой информации в ИСПР

в виде универсального встраиваемого программного модуля с возможностью гибкой настройки всех этапов нечеткого логического вывода. Использование объектно-ориентированной методологии проектирования ПО [4, 5] предоставляет разработчику значительные преимущества на этапах его создания, отладки, внедрения и сопровождения за счет декомпозиции системы на отдельные объекты, отделения описания внутреннего устройства объектов от протоколов взаимодействия между ними (инкапсуляция), построения иерархии объектов (наследование), повторного использования кода, реализации особенностей поведения близких по функциональности объектов (полиморфизм).

На рис.1 приведены основные классы разработанной с использованием объектно-ориентированной методологии библиотеки, которые отражают ключевые сущности, используемые при выполнении нечеткого логического вывода: лингвистические переменные (Var), входные (InputVar) и выходные (OutputVar) переменные; термы лингвистических переменных (Term); функции принадлежности термов (MemberFunc – класс-родитель в иерархии классов функций принадлежности), правила выполнения нечеткого логического вывода (Rule). Также на рис.1 представлены дополнительные классы: FSystem – система нечеткого логического вывода, именно с этим классом и будет реализовываться пользовательский интерфейс; Fuzzy – вспомогательный статический класс, в котором заложена логика реализации всех этапов нечеткого логического вывода; FuzzySet – нечеткое множество (применяется для хранения результата).

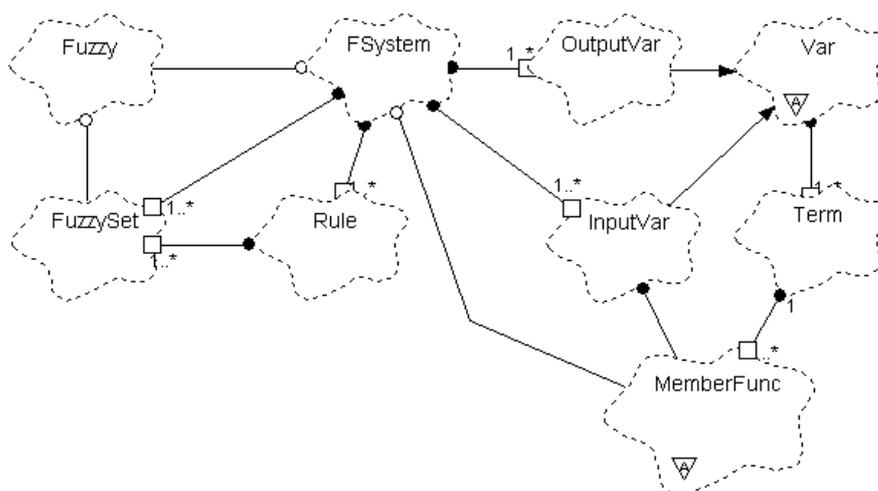


Рис. 1 Диаграмма основных классов библиотеки:

- |  |                                    |  |                          |
|--|------------------------------------|--|--------------------------|
|  | - абстрактный класс;               |  | - агрегация по ссылке;   |
|  | - отношение использования;         |  | - мощность связи (1..n); |
|  | - отношение агрегации (включения); |  | - отношение наследования |

При разработке данного ПО большое внимание уделено универсальности библиотеки классов: программный код написан на языке C# под платформу .NET Framework, обеспечивая аппаратную независимость, возможность использования на различных платформах (при наличии установленной .NET Framework). Широкий набор настроек позволяет применять библиотеку для решения разноплановых задач при проектировании нечетких ИСПР, в частности, как при выборе соответствующего алгоритма нечеткого логического вывода (Mamdani, Sugeno, Tsukamoto, Singleton, алгоритм классификации) [5], так и при выборе формы функций принадлежности для нечетких термов (21 вид).

В библиотеке классов предусмотрены также различные способы реализации каждого из этапов нечеткого логического вывода [6] (рис. 2): фаззификации (для четких и нечетких входных значений); агрегации, активации, аккумуляции (всего 50 видов); импликация (23 вида); дефаззификации (6 видов). Возможность проектирования иерархических многоуровневых систем нечеткого логического вывода при помощи классов создан-

ной библиотеки является весьма важным преимуществом разработанного специализированного ПО.

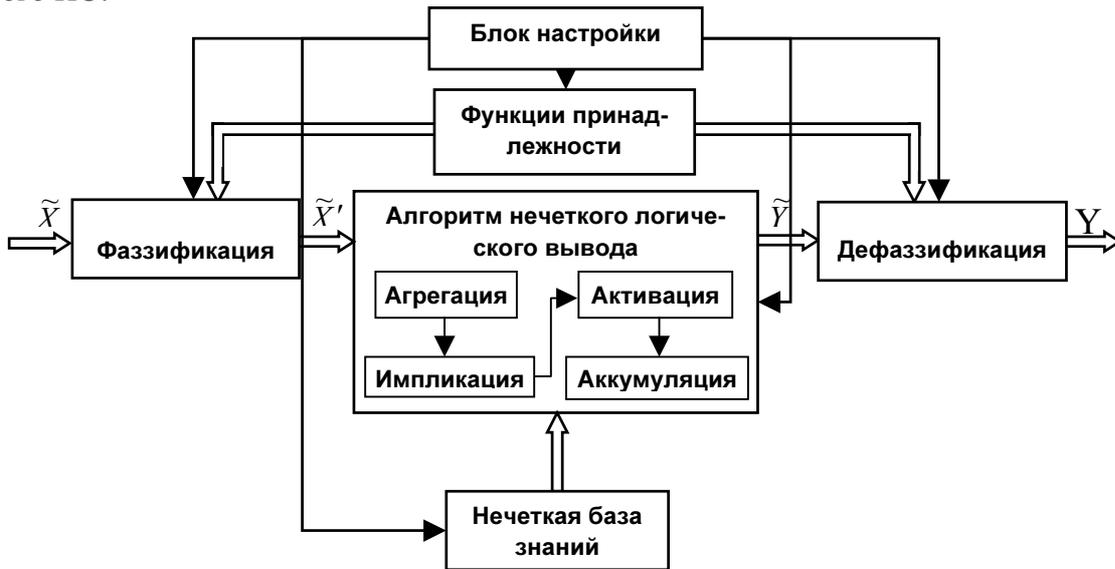


Рис.2. Взаимосвязи компонентов в структуре нечеткого логического вывода:

$\tilde{X}$  – входной вектор (четкие или нечеткие значения);  $\tilde{X}'$  – вектор параметров нечетких множеств, соответствующих входному вектору  $\tilde{X}$ ;  $\tilde{Y}$  – результат логического вывода в виде вектора нечетких множеств;  $Y$  – выходной (четкий) вектор

Гибкость настроек представленной библиотеки классов также обеспечивает широкие возможности для оптимизации нечеткого логического вывода в ИСПР. Преимущества использования библиотеки классов продемонстрируем на примере проектирования простейшего ПД-регулятора (с минимально возможным числом лингвистических термов входных и выходных переменных) для нескольких типов объектов управления (рис. 3), в частности:

интегратора с апериодическим звеном 
$$W(p) = \frac{K}{p(Tp + 1)}, \tag{1}$$

колебательного звена 
$$W(p) = \frac{K}{T^2 p^2 + 2T\xi p + 1}, \tag{2}$$

объекта управления с запаздыванием 
$$W(p) = \frac{Ke^{-\tau p}}{p(Tp + 1)}, \tag{3}$$

где  $K$  – коэффициент передачи;  $T$  – постоянная времени;  $\xi$  – декремент затухания;  $\tau$  – время запаздывания.

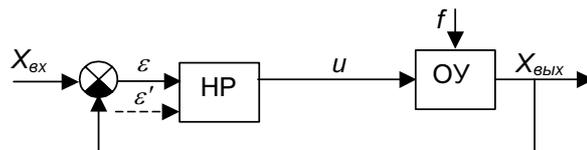


Рис. 3 – Структура системы управления с нечетким регулятором:

$X_{вх}$  – входной сигнал;  $X_{вых}$  – выходной сигнал;  $\varepsilon$  - отклонение;  $\varepsilon'$  – производная от отклонения;  $u$  - управляющий сигнал;  $f$  - возмущающее воздействие; НР – нечеткий регулятор; ОУ - объект управления

Для реализации нечеткого регулятора остановимся на алгоритме нечеткого логического вывода типа Mamdani. Входными переменными для нечеткого ПД-регулятора являются отклонение ( $\varepsilon$ ) выходного сигнала  $X_{вых}$  от установленного на входе значения  $X_{вх}$  (положительное, отрицательное) и производная ( $\varepsilon'$ ) от этого отклонения (положительная, отрицательная). Переменная  $u$  представляет собой уровень значения управляющего сигнала, формируемого на выходе нечеткого регулятора (положительное, нулевое, отрицательное). На рис. 4 приведены термы соответствующих лингвистических переменных НР.

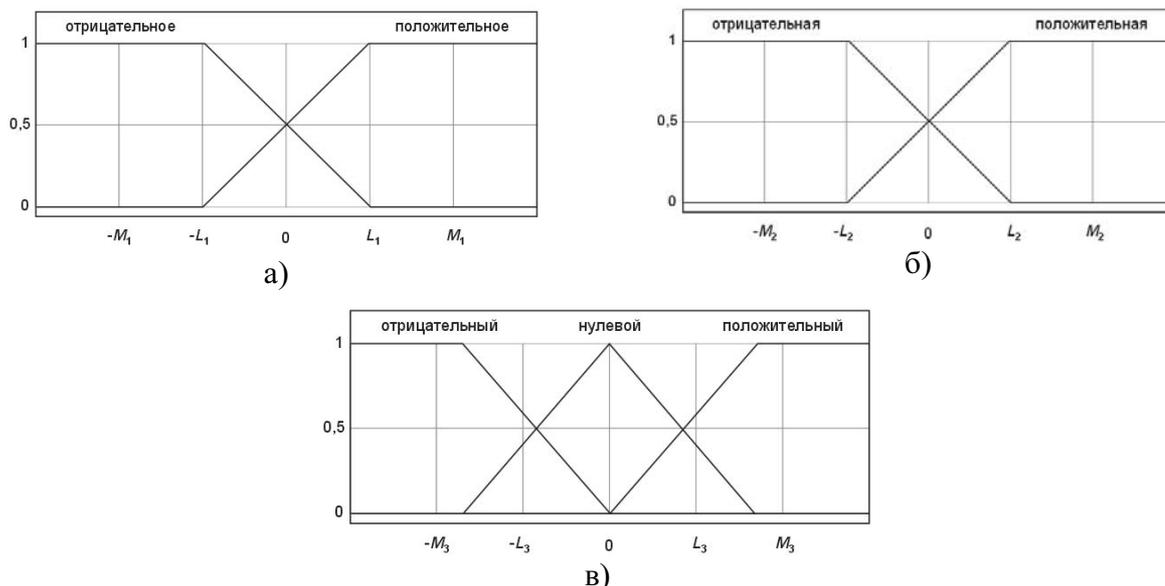


Рис. 4 – Термы лингвистических переменных:

а) переменной «отклонение»; б) переменной «производная от отклонения»; в) переменной «управляющий сигнал»

Приведенная в таблице 1 база нечетких правил отражает зависимости между входными и выходной лингвистическими переменными НР.

Таблица 1

База нечетких правил НР

Значение $\varepsilon$	Значение $\varepsilon'$	Значение $u$
положительное	положительное	положительное
положительное	отрицательное	нулевое
отрицательное	положительное	нулевое
отрицательное	отрицательное	отрицательное

Для реализации процессов регулирования с НР сформированы следующие рекуррентные дискретные модели объектов управления (1)-(3):

- для объекта управления (1):

$$x[n + 2] = x[n + 1]C_1 + x[n]C_2 + u[n]d_0, \tag{4}$$

где  $C_1 = 2 - \frac{h}{T}$ ;  $C_2 = \frac{h}{T} - 1$ ;  $d_0 = \frac{K \cdot h^2}{T}$ ;

- для объекта управления (2) рекуррентная модель также имеет вид (4),

где  $C_1 = 2 \cdot \left(1 - \frac{\xi \cdot h}{T}\right)$ ;  $C_2 = \frac{2 \cdot \xi \cdot h}{T} - 1 - \frac{h^2}{T^2}$ ;  $d_0 = \frac{K \cdot h^2}{T^2}$ ;

- для объекта управления (3):

$$x[n+2] = x[n+1]C_1 + x[n]C_2 + u[n-k]d_0, \quad (5)$$

где  $C_1 = 2 - \frac{h}{T}$ ;  $C_2 = \frac{h}{T} - 1$ ;  $d_0 = \frac{K \cdot h^2}{T}$ ;  $k = \frac{\tau}{h}$ .

В приведенных дискретных моделях использованы следующие обозначения:  $h$  – шаг дискретизации ( $\Delta t = h = 0,01$ с);  $u[n]$  - управляющий сигнал на  $n$ -м итерационном шаге (в нечетком регуляторе  $u[n]$  формируется посредством нечеткого логического вывода, а в классическом ПД-регуляторе [7] – как  $u[n] = k_p \cdot \varepsilon + k_D \cdot \varepsilon'$ , где  $k_p, k_D$  - коэффициенты настройки ПД-регулятора).

Структурно-параметрическая оптимизация нечеткого ПД-регулятора осуществлена с использованием различных настроек, предоставляемых разработанной на основе объектно-ориентированных технологий библиотекой классов. При этом решались задачи оптимального выбора методов алгоритмической реализации всех этапов нечеткого логического вывода и нахождения оптимальных параметров лингвистических термов входных и выходной переменных. Результаты структурно-параметрической оптимизации НР приведены в таблицах 2-4.

Таблица 2

Результаты структурно-параметрической оптимизации нечеткого регулятора для объекта управления (1)

Этап нечеткого логического вывода	Метод реализации	Формула
Агрегация	t-норма Minimum	$T(a, b) = \min(a, b)$
Импликация	Gödel	$I(a, b) = \begin{cases} 1, & \text{если } a \leq b \\ b, & \text{иначе} \end{cases}$
Активация/аккумуляция	t-норма Minimum	$T(a, b) = \min(a, b)$
Дефаззификация	Bisector	-
Параметры термов нечетких лингвистических переменных		
$L_1 = 1; L_2 = 3; L_3 = 1; M_1 = M_2 = M_3 = 3$		

Таблица 3

Результаты структурно-параметрической оптимизации нечеткого регулятора для объекта управления (2)

Этап нечеткого логического вывода	Метод реализации	Формула
Агрегация	t-норма Nilpotent	$T(a, b) = \begin{cases} \min(a, b), & \text{если } (a+b) > 1 \\ 0, & \text{иначе} \end{cases}$
Импликация	Zadeh	$I(a, b) = \begin{cases} 1, & \text{если } a \leq b \\ \frac{b}{a+(1-b)}, & \text{иначе} \end{cases}$
Активация/аккумуляция	среднее Geometric	$M(a, b) = (a \cdot b)^{1/2}$
Дефаззификация	Bisector (медиана)	-
Параметры термов нечетких лингвистических переменных		
$L_1 = 1; L_2 = 3; L_3 = 1; M_1 = M_2 = M_3 = 3$		

Результаты структурно-параметрической оптимизации нечеткого регулятора для объекта управления (3)

Этап нечеткого логического вывода	Метод реализации	Формула
Агрегация	t-норма Minimum	$T(a, b) = \min(a, b)$
Импликация	Gödel	$I(a, b) = \begin{cases} 1, & \text{если } a \leq b \\ b, & \text{иначе} \end{cases}$
Активация/аккумуляция	среднее Geometric	$M(a, b) = (a \cdot b)^{1/2}$
Дефаззификация	Weight center (центр масс)	-
Параметры термов нечетких лингвистических переменных		
$L_1 = 1; L_2 = 3; L_3 = 1; M_1 = M_2 = M_3 = 3$		

Для оценки эффективности процессов оптимизации нечеткого ПД-регулятора сравним его с классическим ПД-регулятором с оптимальными коэффициентами настроек.

Для оптимизации классического ПД-регулятора необходимо прежде всего сформировать желаемый переходный процесс (рис. 5), параметры которого приведены в таблице 5, а в дальнейшем выполнить параметрическую оптимизацию соответствующих коэффициентов пропорциональной и дифференциальной составляющих.

Таблица 5

Требуемые параметры желаемого переходного процесса

Наименование параметра	Единица измерения	Значение
Время регулирования	с	2
Время нарастания переходного процесса	с	1,5
Статическое отклонение (относительно уровня входного сигнала)	%	5
Величина выходного сигнала в момент первого достижения установленного значения	%	95
Перерегулирование (относительно установившегося уровня выходного сигнала)	%	1
Отрицательное перерегулирование (относительно установившегося уровня выходного сигнала)	%	1

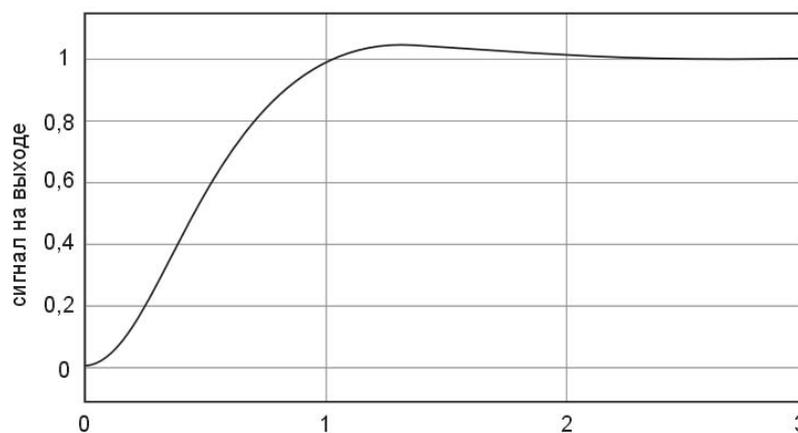


Рис. 5 Желаемый переходный процесс для оптимизации параметров ПД-регулятора

Результаты оптимизации параметров традиционных ПД-регуляторов

Модель объекта управления	Настройка $k_P$ для пропорциональной составляющей	Настройка $k_D$ для дифференциальной составляющей
Согласно (1) при $K = 10, T = 1$	1,3697	0,5940
Согласно (2) при $K = 2, T = 1, \xi = 0,25$	10	3,8882
Согласно (3) при $K = 1, T = 1, \tau = 0,2$	4,7774	2,8279

Сравнительный анализ эффективности нечетких и соответствующих классических ПД-регуляторов легко осуществить на основе переходных процессов, приведенных на рис.6-8.

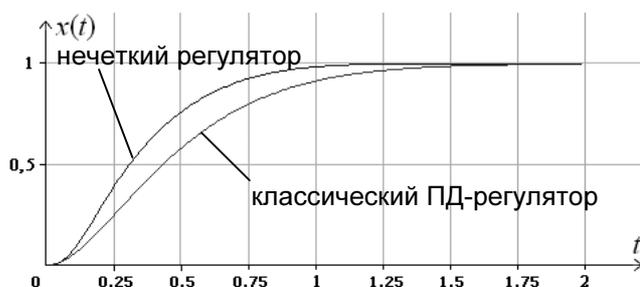


Рис. 6 Сравнение классического и нечеткого ПД-регуляторов для объекта управления (1)

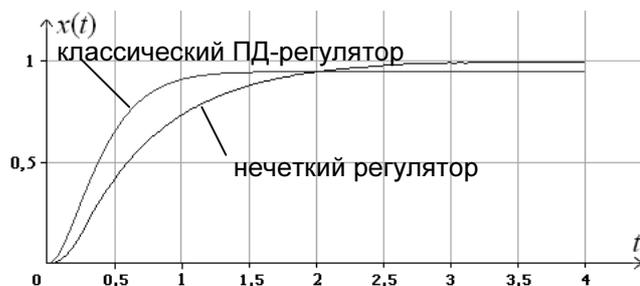


Рис. 7 Сравнение классического и нечеткого ПД-регуляторов для объекта управления (2)

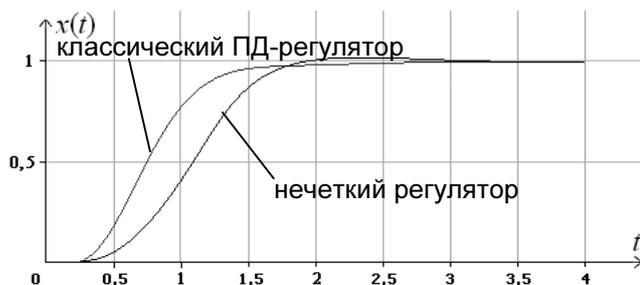


Рис. 8 Сравнение классического и нечеткого ПД-регуляторов для объекта управления (3)

Сравнительный анализ показывает, что использование нечетких и классических ПД-регуляторов после их оптимальной настройки дают достаточно хорошие и близкие

результаты. Вместе с тем, для объекта управления (1) нечеткий регулятор обеспечивает более высокое быстродействие, для объекта управления (2) – меньшую статическую ошибку, а для объекта управления (3) – меньшую длительность времени нарастания сигнала.

**Выводы.** Результаты моделирования подтверждают высокую эффективность использования разработанной библиотеки классов при синтезе и структурно-параметрической оптимизации нечетких регуляторов для различных объектов управления. Показатели качества переходных процессов при нечетком управлении с использованием синтезированного НР не уступают показателям качества соответствующих классических регуляторов, а в некоторых случаях и превосходят их. При этом следует отметить, что преимуществом НР при их аппаратной реализации является отсутствие динамических составляющих, а значит зависимости между возможными значениями входов и выхода НР могут быть предварительно рассчитаны на стадии проектирования, а на основе характеристических поверхностей НР реализованы с помощью табличного спецпроцессора [8].

Разработанная библиотека классов дает широкие возможности при проектировании нечетких регуляторов, а также при осуществлении их структурно-параметрической оптимизации. В дальнейшем целесообразно проведение исследований по синтезу и оптимизации нечетких ПИД-регуляторов на основе библиотеки классов, созданию библиотеки объектов управления различных типов и настройке регуляторов для эффективного управления такими объектами в условиях неопределенности.

This paper covers the issues of designing fuzzy information processing class library as a universal programming module which allows tuning all of the fuzzy logic inference stages. The functionality of the library is illustrated while synthesizing and optimizing fuzzy PD-regulator in comparison with classic PD-regulator.

1. Заде Л.А. Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии информационных/интеллектуальных систем. – Новости Искусственного Интеллекта, №2-3, 2001, с. 7 – 11.
2. Кондратенко Ю.П. Интеллектуальные системы поддержки принятия решений при управлении судами в экстремальных условиях – Судходство, №3, 2004. – с. 21.
3. Кондратенко Ю.П., Мухортова К.В. Обобщенная библиотека классов для обработки нечеткой информации // Материалы научно-практической конференции "Информационные технологии - в науку и образование". - Харьков, 2005. - с. 100 - 101
4. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд. – Москва: Издательство Бином, СПб: Невский диалект, 2001. – 560 с.
5. Кондратенко Г. В. Об'єктна декомпозиція задачі оптимального планування маршрутів // Матеріали науково-практичної конференції. Стан і перспективи розвитку новітніх науково-освітніх комп'ютерних технологій. – Миколаїв: Видавництво МДГУ ім. П. Могили, 2003. – с. 98–100
6. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. – Москва: Горячая линия – Телеком, 2004. – 452 с.
7. Дорф Р. Современные системы управления / Р. Дорф, Р. Бишоп; Пер с англ. Б. И. Копылова. – Москва: Лаборатория Базовых Знаний, 2004. – 832 с.: илл.
8. Кондратенко Ю. П., Сидоренко С. А. Методи синтезу нечітких контролерів для підвищення швидкодії та точності процесів формування управляючих сигналів // Збірник наукових праць УДМУ. – Миколаїв. – 2002. – Вип. 1 (379). – с. 121-129