

УДК 62-50:004.942

**Н.А. Софин, Ю.Н. Хижняков**

**N.A. Sofin, Yu.N. Khizhnyakov**

Пермский национальный исследовательский  
политехнический университет

Perm National Research Polytechnic University

## **ПРИМЕНЕНИЕ АЛГОРИТМОВ НЕЧЕТКОГО УПРАВЛЕНИЯ ДЛЯ СТАБИЛИЗАЦИИ НЕУСТОЙЧИВЫХ ОБЪЕКТОВ**

### **APPLICATION OF ALGORITHMS OF FUZZY CONTROL FOR STABILIZATION OF UNSTABLE OBJECTS**

Регуляторы на основе нечеткой логики позволяют управлять объектами, математическое описание которых является очень сложным. В качестве примера такого объекта рассматривается физическая модель перевернутого маятника. Данная модель позволяет отлаживать реализации нечеткого регулятора, которые могут быть основаны на логике Заде, Мамдани, В.И. Гостева, Ларсена.

**Ключевые слова:** перевернутый маятник, контроллер Arduino Uno, программирование, нечеткая логика, терм-множество, лингвистические переменные.

Controllers based on fuzzy logic allows control objects, which has difficult mathematics description. For example it, considered physical model inverted pendulum. It allows debug difference implementations fuzzy controller algorithm, which based on Zade, Mamdani, V.I. Gostev, Larsen logic.

**Keywords:** inverted pendulum, controller Arduino Uno, programming, fuzzy logic, term set, linguistic variables

Известны случаи, когда выход из строя автоматической системы управления неустойчивым объектом представлял собой существенную опасность для человека и окружающей среды (к примеру, авария на Чернобыльской АЭС или крушение самолета). Начинать изучение методов управления неустойчивыми объектами целесообразно на простых примерах, одним из которых является классический перевернутый маятник, внешний вид которого изображен на рис. 1.

Этот стенд представляет собой полноприводную платформу, состоящую из следующих компонентов:

- четырех двигателей постоянного тока, максимально допустимое напряжение которых составляет 18 В;
- контроллера Arduino Uno в качестве аппаратной основы;

- Bluetooth-модуля HC-05, необходимого для функционирования мобильной системы управления на базе операционной системы Android;
- LCD Keypad Shield в качестве встроенной системы управления;
- встроенных разделенных источников питания на 18 (для силовой части) и 9 В (для контроллера).

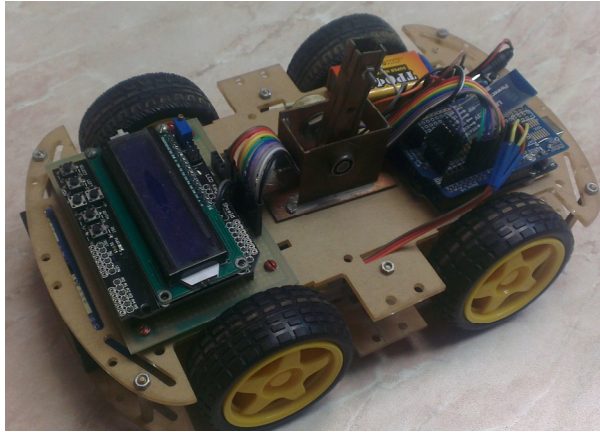


Рис. 1. Внешний вид лабораторного стенда

Суть всех предыдущих модернизаций состояла в улучшении аппаратной части и систем управления. В данной статье рассматривается последний этап модернизации, основной задачей которой является переход на нечеткий алгоритм стабилизации объекта [1]. Нечеткая логика полезна для тех объектов, математическое описание которых является слишком сложным. Регуляторы на основе нечеткой логики похожи на совокупность правил, составленных экспертами и формализованных в нотации того или иного языка программирования.

Структурная схема лабораторного стенда с нечетким регулятором представлена на рис. 2.

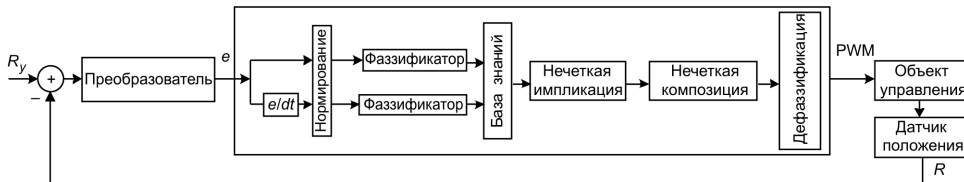


Рис. 2. Структурная схема лабораторного стенда

На вход системы поступает сигнал с датчика  $R_T$  и значение уставки  $R_y$  в единицах сопротивления. Обычно нечеткие регуляторы работают с такими понятиями (лингвистическими переменными (ЛП)), как отклонение и произ-

водная отклонения какого-либо сигнала, поэтому в блоке преобразования сигнал превращается в отклонение, измеряемое в градусах. Сигнал отклонения и его производная поступают на вход блока нормирования, суть которого заключается в том, чтобы превратить диапазон изменения отклонения и его производной в нормированный диапазон от  $-1$  до  $1$ . Делается это с использованием уравнения прямой, проходящей через две точки:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}. \quad (1)$$

К примеру, отклонение изменяется в диапазоне от  $-5$  до  $5^\circ$ . Следовательно, нормированное значение отклонения будет рассчитываться по следующей формуле:

$$y = \frac{xy_2 - x_1y_2 + y_1x_2}{x_2} \quad (2)$$

при  $x_1 = -5, y_1 = -1, x_2 = 5, y_2 = 1 \Rightarrow y = 0,2x$ .

Далее идет процесс фаззификации, т.е. превращение четкого значения в нечеткое. Под нечетким значением нужно представлять одно из таких понятий (терм), как:

- положительная большая (ПБ) типа Z;
- положительная средняя (ПС) типа L;
- положительная малая (ПМ) типа L;
- норма типа L;
- отрицательная малая (ОМ) типа L;
- отрицательная средняя (ОС) типа L;
- отрицательная большая (ОБ) типа S.

В совокупности термы образуют терм-множество лингвистической переменной. Математическое описание этих терм представлено на рис. 3.

Есть еще одна терма под названием синглетон, назначение которой состоит в том, чтобы постоянно определять значения функций принадлежности (ФП) по формулам на рис. 3. Работа фаззификатора как для отклонения, так и для его производной изображена на рис. 4.

Результатом фаззификации для каждой ЛП являются две термы и соответствующие им значения ФП. Полученный блок данных поступает на обработку в базу знаний, которая составляется экспертами. В общем случае база знаний представляет собой коллекцию правил в формате «если-то». Правила могут активироваться в количестве от 1 до 4. Результатом правила также является терма, но уже типа синглетон, и она принадлежит терм-множеству выходной ЛП (это ШИМ-сигнал). Работа базы знаний представлена на рис. 4.

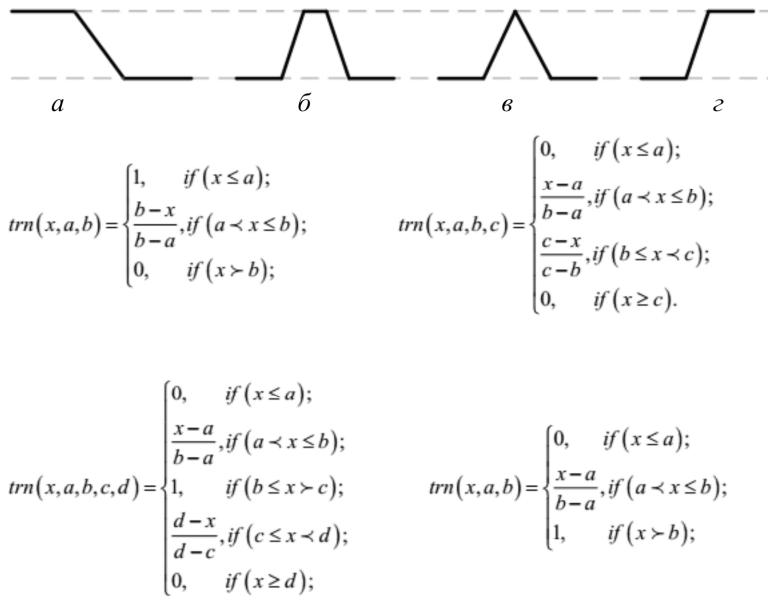


Рис. 3. Математическое описание линейных терм:  
 $a, б, г$  – трапецидального тела;  $в$  – треугольного тела

Затем начинается работа блока нечеткого логического вывода, который состоит из блока нечеткой импликации и блока нечеткой композиции. Нечеткая импликация из результирующей выборки базы данных выбирает одинаковые термы, затем среди них ищется такая терма, которая соответствует минимальному значению ФП. В отличие от импликации, композиция занимается поиском термы, которой соответствует максимальное значение ФП.

Последним блоком является блок дефаззификации, задача которого превращение нечеткого значения в четкое, т.е. в значение ШИМ-сигнала. Происходит это по следующей формуле:

$$\text{ШИМ} = \frac{\sum x \mu_a}{\sum \mu_a}, \quad (3)$$

где  $x$  – значение ШИМ-сигнала, которое соответствует терме выходной ЛП (эти значения определяются проектировщиком);  $\mu_a$  – значение ФП. Графическая интерпретация блоков импликации, композиции и дефаззификации изображена на рис. 4.

На рис. 5 представлена демонстрационная версия программы мониторинга работы нечеткого регулятора. Эта программа сформирована с использованием специально разработанного фреймворка [2] на языке С# [3, 4]. Исходный код разработанного регулятора в данной статье не приводится в связи с его большим объемом.

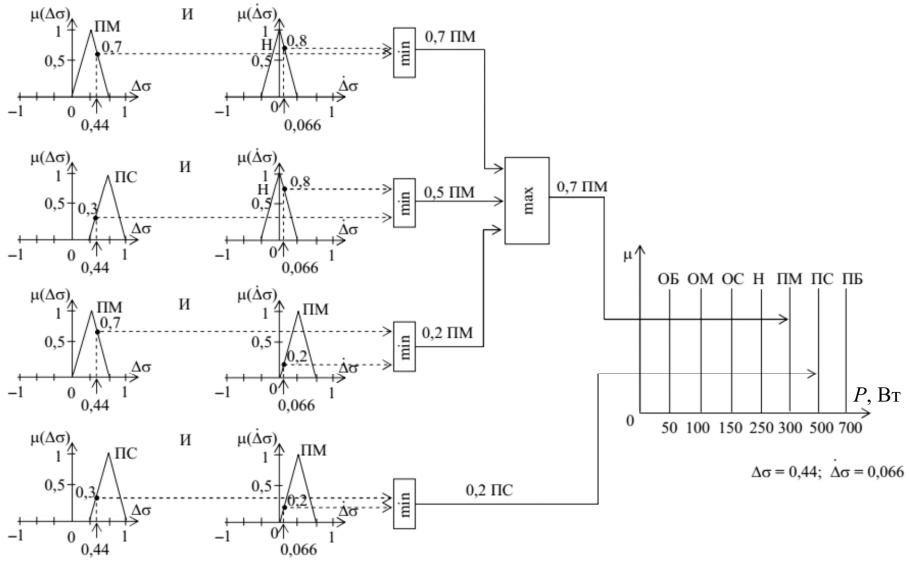


Рис. 4. Графическая интерпретация блоков импликации, композиции и дефазификации

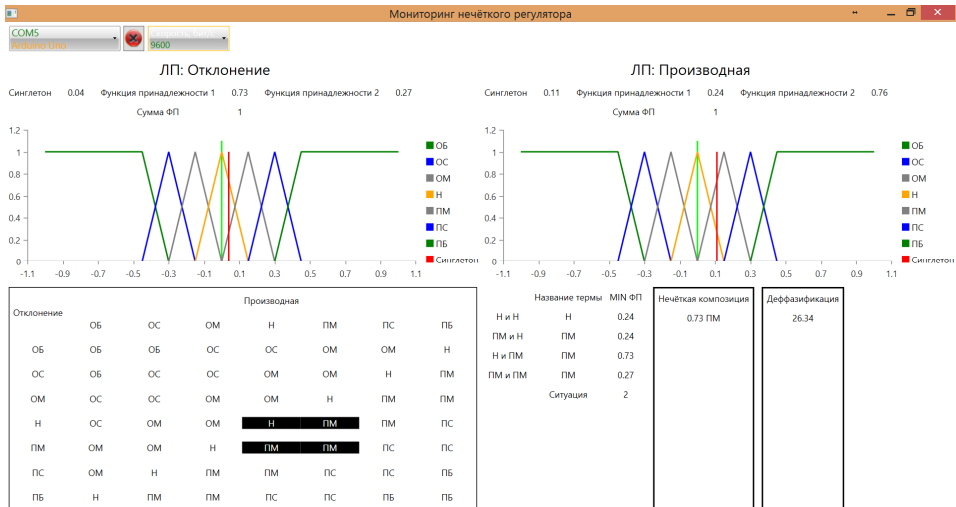


Рис. 5. Внешний вид программы

Реализация алгоритма нечеткой логики создает основу для понимания работы более сложных алгоритмов управления, основанных на нейро-нечетких регуляторах. Среди направлений дальнейшей модернизации можно выделить разработку и тестирование нейро-нечеткого регулятора, изучение аппарата нейронных сетей.

### Список литературы

1. Хижняков Ю.Н. Нечеткое, нейронное и гибридное управление. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2013. – 303 с.
2. Софин Н.А. Создание фреймворка на языке C# для разработки программного обеспечения верхнего уровня систем управления // Автоматизированные системы управления и информационные технологии. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2015. – С. 217–222.
3. Ватсон Б. C# 4.0 на примерах. – СПб., 2011. – 590 с.
4. Мак-Дональд М. WPF. Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0. – М.; СПб.; Киев, 2013. – 1015 с.

Получено 08.09.2016

**Софин Николай Александрович** – магистр кафедры «Автоматика и телемеханика», электротехнический факультет, Пермский национальный исследовательский политехнический университет, e-mail: skkult@yandex.ru.

**Хижняков Юрий Николаевич** – доктор технических наук, профессор кафедры «Автоматика и телемеханика», электротехнический факультет, Пермский национальный исследовательский политехнический университет.