

ПРИМЕНЕНИЕ НЕЙРО-НЕЧЕТКОЙ ЛОГИКИ В ЗАДАЧЕ ФОРМИРОВАНИЯ ТЕМПЕРАТУРНОЙ УСТАВКИ В ИНКУБАЦИОННОМ ПРОЦЕССЕ

Дуванов Е.С., Шишов М.Ю.

Липецкий государственный технический университет, Липецк, Россия
evgenyduvanov@yandex.ru, fandimfromitaly@yandex.ru

Пашченко А.Ф.

Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
paschenko_alex@mail.ru

Аннотация. В работе рассмотрено применение нейро-нечеткой логики в задаче формирования температурных уставок в инкубационном процессе. Обоснован выбор функций принадлежности и число нечетких правил, описывающих технологический процесс. Представлен разработанный специализированный программный, реализованный средствами языка программирования Python, и продемонстрировано его применение в производственных условиях.

Ключевые слова: нейро-нечеткая логика, инкубационный процесс, температурная уставка, ANFIS, Python.

Введение

Одной из распространенных проблем при инкубации куриных яиц является нарушение оптимальных условий – особенно переохлаждение или перегрев эмбрионов [1]. Поддержание стабильной температуры и влажности алгоритм позволяет автоматизированной системе управления инкубационными камерами адаптироваться к изменениям условий и поддерживать оптимальные параметры инкубации [1-6].

Нейро-нечеткий алгоритм представляет собой гибридный подход, сочетающий в себе гибкость нейронных сетей и точность нечеткой логики. Данная комбинация позволяет создавать интеллектуальные системы управления, обладающие рядом ключевых преимуществ для оптимизации процессов инкубации [7]:

- оперативно реагировать на изменение температуры и влажности в инкубационных камерах,
- автоматически корректировать режимы работы оборудования для поддержания оптимальных технологических условий микроклимата,
- прогнозировать возможные отклонения параметров и формировать превентивные экспертные решения.

Таким образом, применение нейро-нечеткого алгоритма для формирования температурной уставки позволяет существенно повысить стабильность и качество инкубации куриных яиц благодаря точному определению опорного значения температуры внутри допустимого диапазона, снижая риски гибели эмбрионов и способствуя увеличению выхода здорового молодняка.

1. Выбор нечеткой модели для прогнозирования уставки

В конце 1960 – начала 1970-х гг. Лотфи Заде сформулировал концепцию нечетких множеств и логики – фундамент для нейро-нечетких алгоритмов. Вклад Заде, а также исследований Мамдани и Беллмана заложил теоретическую основу для моделирования нечеткости и человеческого принятия решений [8, 9].

В 1970–1980-х гг. появились первые практические реализации: нейро-нечеткие системы управления, которые эффективно обрабатывали неточные данные. Одновременно внедрялись экспертные системы для медицины и экономики: гибкость нечеткой логики оказалась полезной при работе с неоднозначной информацией. Нейро-нечеткие подходы продолжают эволюционировать и применяются в разных сферах, сочетая адаптивность нейросетей с логикой нечеткости. Особенno важна модель *ANFIS* (*Adaptive Neuro-Fuzzy Inference System*), объединяющая преимущества обоих подходов [7].

Предположим, что адаптивная система нейро-нечеткого вывода *ANFIS* имеет два входа x_1, x_2 и один выход y . Соответствующая нечеткая модель Сугено с двумя правилами запишется как

$$R^1: \text{если } x_1 \text{ есть } X_1^1 \text{ и } x_2 \text{ есть } X_2^1, \text{ то } y = c_0^1 + c_1^1 x_1 + c_2^1 x_2. \quad (1)$$

$$R^2: \text{если } x_1 \text{ есть } X_1^2 \text{ и } x_2 \text{ есть } X_2^2, \text{ то } y = c_0^2 + c_1^2 x_1 + c_2^2 x_2. \quad (2)$$

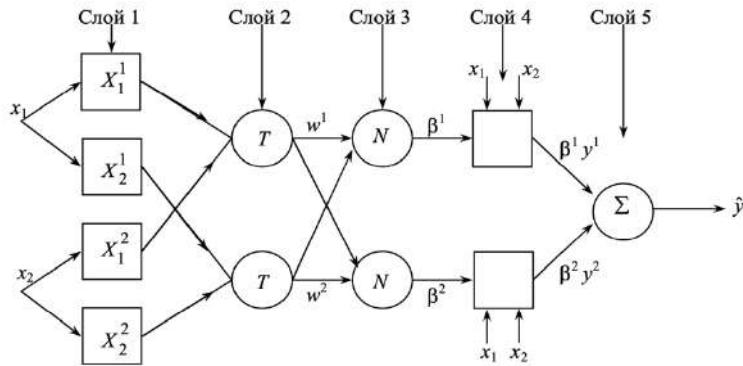


Рис. 1. Архитектура ANFIS

На рис. 1 представлена архитектура ANFIS, состоящая из 5 слоев, в которых реализуется механизм вывода решения [7, 10, 11].

Слой 1. Вычисляет степени принадлежности, т.е. значения функций принадлежности $X_i^\theta(x_i)$ при заданных значениях входов x_1 и x_2 .

Слой 2. Выводит результат произведения входных сигналов, именуемый истинностью правила

$$w^\theta = X_1^\theta(x_2) \cdot X_2^\theta(x_2), \theta = 1,2. \quad (3)$$

Слой 3. Находит нормированное значение истинности правила или нечеткой функции

$$\beta^\theta = w^\theta / (w^1 + w^2), \theta = 1,2. \quad (4)$$

Слой 4. Вычисляет параметры следствия

$$\beta^\theta y^\theta = \beta^\theta (c_0^\theta + c_1^\theta x_1 + c_2^\theta x_2), \theta = 1,2. \quad (5)$$

Слой 5. Суммирует все поступающие сигналы

$$\hat{y} = \sum_{\theta=1}^2 \beta^\theta y^\theta = \frac{\sum_{\theta=1}^2 w^\theta y^\theta}{\sum_{\theta=1}^2 w^\theta}. \quad (6)$$

В качестве функций принадлежности, задающих нечеткие множества, используется колокольная функция принадлежности.

$$X(x) = (1 + ((x - d_1)/d_2)^{2d_3})^{-1}. \quad (7)$$

Модель ANFIS обучается путем настройки параметров нечетких правил и весов нейронной сети с использованием методов оптимизации, таких как градиентный спуск или алгоритм обратного распространения ошибки. Это позволяет модели ANFIS адаптироваться к данным и выполнять сложные задачи прогнозирования и классификации [11].

2. Разработка экспертной системы

При разработке программного обеспечения нейро-нечеткой системы управления выбор сделан в пользу языка программирования *Python*, что обусловлено его многими преимуществами. Прежде всего, данный язык располагает обширным набором библиотек и инструментов для реализации интеллектуальных экспертных систем, в том числе на основе нечеткой логики и нейронных сетей, что позволяет эффективно внедрять алгоритмы обработки данных, машинного обучения и нечеткого вывода. Более того, интегрируется с промышленными контроллерами, такими как *Siemens SIMATIC S7-1200*. Взаимодействие с программируемыми логическими промышленными контроллерами возможно через библиотеки вроде *python-snap7* или *S7comm*, обеспечивающие обмен данными по *Ethernet* с использованием протокола *Siemens S7*, позволяя в реальном времени считывать значения из *DB*-блоков, обрабатывать их и записывать обратно, обеспечивая полноценный мониторинг и управление аппаратной частью объекта [12-13].

2.1. Экспертная система для температурных уставок

Процесс промышленной инкубации яиц специалисты разделяют на две основные стадии – инкубацию и вывод цыплят [1, 3-6]. Исследования показали, что инкубацию можно разбить на пять периодов, в каждом из которых требуется поддерживать оптимальную температуру.

Таблица 1. Функции принадлежности программного комплекса

Нечеткая переменная	Функция принадлежности	Тип функции принадлежности	Параметры
Q	Q ₁	zmf	[0,07433 0,3246]
	Q ₂	gbellmf	[0,146 2,54 0,3508]
	Q ₃	gbellmf	[0,641 2,47 1,328]
	Q ₄	gbellmf	[2,58 2,5 4,558]
	Q ₅	smf	[4,286 9,97]
H	H ₁	smf	[51,55 62,8]
	H ₂₋₃	gbellmf	[8 3,01 49,56]
	H ₄₋₅	zmf	[38,1 44,9]
T_C	T ₁ _C	smf	[37,58 38]
	T ₂₋₃ _C	gbellmf	[0,1717 2,61 37,6]
	T ₄ _C	gbellmf	[0,2113 37,2]
	T ₅ _C	zmf	[36,68 37,32]
T	T ₁	smf	[37,58 38]
	T ₂₋₃	gbellmf	[0,1717 2,61 37,6]
	T ₄	gbellmf	[0,2113 37,2]
	T ₅	zmf	[36,68 37,32]

Уставочная температура (T) определяется сочетанием трех параметров: количество теплоты, выделяемого одним яйцом (Q), текущей температуры (T_C) в инкубационном шкафу и уровня влажности (H).

Экспертная система на основе этих данных формирует температурную уставку для поддержания оптимального микроклимата. На основании знаний экспертов [1] подобраны функции принадлежности (табл. 1), характеризующие каждый из этапов инкубации, и сформулированы нечеткие правила (8), позволяющие гибко адаптировать управление тепловым режимом. В качестве функций принадлежности выбраны колоколообразные ($gbellmf$), а также Z-образная (zmf) и S-образная (smf). Эти функции применяются для задания мягких, нечетких границ между лингвистическими термами.

Важно отметить, что объединение функций принадлежности, описывающих температурные значения второго и третьего этапов инкубации, проводится с целью оптимизации вычислений программного кода и учитывается в формулировке базы правил, позволяя повысить точность прогнозирования результатов [4].

Ниже представлены графики функций принадлежностей, описывающие все параметры управления процесса инкубации (рис. 2).

- R^1 : если q есть Q_1 и h есть H_1 и tc есть T_{1_C} , то t есть T_1 ,
 R^2 : если q есть Q_1 и h есть H_{2-3} и tc есть T_{2-3_C} , то t есть T_{2-3} ,
 R^3 : если q есть Q_1 и h есть H_{4-5} и tc есть T_{4_C} , то t есть T_4 ,
 R^4 : если q есть Q_1 и h есть H_{4-5} и tc есть T_{5_C} , то t есть T_5 ,
 R^5 : если q есть Q_2 и h есть H_1 и tc есть T_{1_C} , то t есть T_1 ,
 R^6 : если q есть Q_2 и h есть H_{2-3} и tc есть T_{2-3_C} , то t есть T_{2-3} ,
 R^7 : если q есть Q_2 и h есть H_{4-5} и tc есть T_{4_C} , то t есть T_4 ,
 R^8 : если q есть Q_2 и h есть H_{4-5} и tc есть T_{5_C} , то t есть T_5 ,
 R^9 : если q есть Q_3 и h есть H_1 и tc есть T_{1_C} , то t есть T_1 ,
 R^{10} : если q есть Q_3 и h есть H_{2-3} и tc есть T_{2-3_C} , то t есть T_{2-3} ,
 R^{11} : если q есть Q_3 и h есть H_{4-5} и tc есть T_{4_C} , то t есть T_4 ,
 R^{12} : если q есть Q_3 и h есть H_{4-5} и tc есть T_{5_C} , то t есть T_5 ,

 R^{20} : если q есть Q_5 и h есть H_{4-5} и tc есть T_{5_C} , то t есть T_5 .

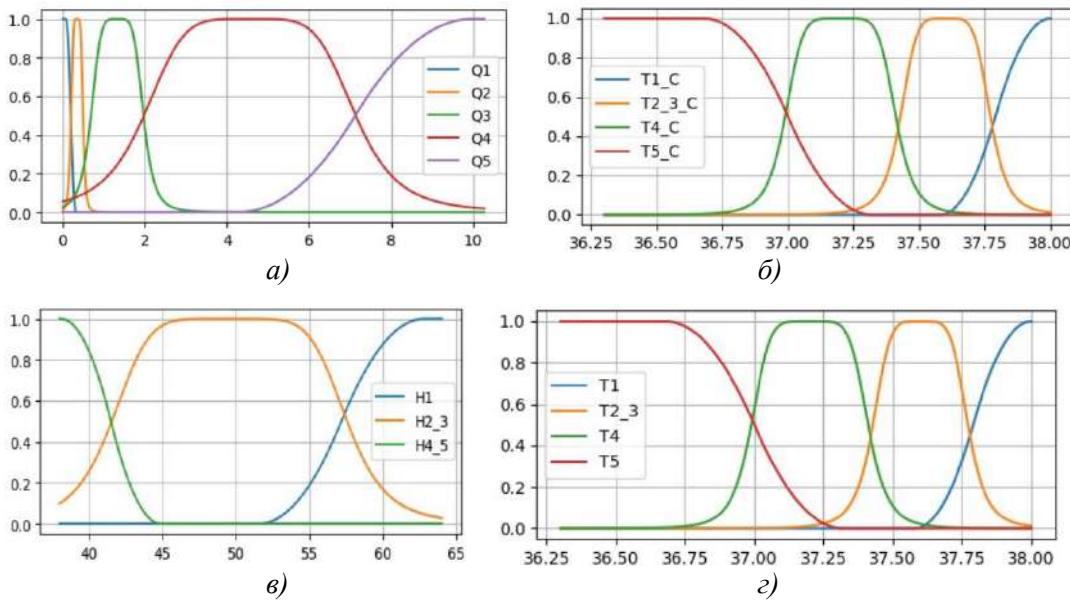


Рис. 2. Функции принадлежности, описывающие а) количество теплоты, б) текущая температура, в) влажность, г) температурная уставка на выходе

2.2. Описание программной реализации экспертной системы

В начальном этапе конфигурации нечеткой системы задаются числовые диапазоны (универсы) для всех входных и выходных переменных. Затем, на основе динамики параметров в течение инкубационного периода, для каждой из входных переменных и для выходной переменной формируется по нескольку функций принадлежности. В библиотеке *skfuzzy.control* классы *ctrl.Antecedent* и *ctrl.Consequent* используются для представления входных и выходных переменных соответственно.

Первый класс (*ctrl.Antecedent*) определяет условия или параметры, которые влияют на работу системы (в данном случае *input 1*, *input 2*, *input 3*), задавая универсы и соответствующие им нечеткие функции принадлежности (*membership functions*).

Второй класс (*ctrl.Consequent*) описывает выходную переменную (*output*), включая ее диапазон значений и функции принадлежности, отражающие возможные реакции системы.

Оба элемента служат структурными компонентами нечеткой логической системы, которая реализует принятие экспертного решения с использованием правил *IF-THEN*. В программе задаются нечеткие правила для всех переменных.

```
rule1 = ctrl.Rule(input1['Q1'] & input2['H1'] & input3['T1_C'], output['T1'])
.....
rule20 = ctrl.Rule(input1['Q5'] & input2['H4_5'] & input3['T5_C'], output['T5'])
```

(9)

В контексте программы класс *ctrl.ControlSystem* предназначен для хранения множества нечетких правил *Rule*, связывающих входные (*Antecedent*) и выходные (*Consequent*) переменные. После заполнения системой правил, на ее основе создается экземпляр *ControlSystemSimulation*, который непосредственно выполняет расчет, принимая на вход конкретные значения переменных.

Используя библиотеку *scikit-fuzzy*, создаем объект нечеткую систему управления и инициализируем ее с помощью функции *ControlSystemSimulation*.

```
anfissimulation = ctrl.ControlSystemSimulation(anfisctrl)
```

(10)

Класс *ctrl.ControlSystemSimulation* библиотеки *skfuzzy.control* предназначен для выполнения вычислений на основе заранее определенной нечеткой системы (*ControlSystem*). Он инициализируется при помощи объекта *ControlSystem* и параметров, таких как *clip_to_bounds* (ограничение входных значений универсом), *cache* (использование кэша для ускорения повторных расчетов) и *flush_after_run* (сброс кеша по окончании определенного числа симуляций).

Основным методом является *compute()*, который обеспечивает полный цикл обработки, фазификация входов, применение правил через методы агрегирования, активации и накопления

(реализуемые в *compute_rule()* согласно инференции Мамдани), а также дефазификация, в результате чего генерируются выходные значения. Метод *reset()* полностью очищает состояния симуляции, возвращая систему в исходное состояние. Дополнительно класс предоставляет метод *print_state()*, который отображает внутренние состояния модели: входные значения, степени принадлежности термов, активации правил и результат дефазификации.

Следующим этапом разработки является создание функции, предназначенной для вычисления температуры в инкубационной камере на основе текущих значений параметров. Функция принимает в качестве входных данных: день, часы, минуты, влажность и текущую температуру. В процессе работы функции осуществляется преобразование значений часов и минут в дни, после чего полученные дни используются для вычисления параметра «выделяемая энергия одним яйцом». Эти данные затем передаются в качестве входных переменных в нечеткую систему управления. На последнем этапе вычисляется выходной параметр – температура в инкубационной камере. Результаты вычислений сохраняются в файл формата *CSV* с использованием соответствующей библиотеки *Python*.

3. Результат работы программы

Разработанный специализированный программный комплекс состоит из нескольких функциональных окон, каждое из которых выполняет определенные задачи [4].

Окно «Уставки» предоставляет пользователю возможность ввода данных для конкретного времени и дня инкубации. Введенные параметры используются для расчета и формирования результатов, которые сохраняются в файле формата *CSV* для дальнейшего анализа данных технологическими специалистами.

Окно «Функции принадлежности» визуализирует текущие нечеткие функции, соответствующие введенным параметрам, включая отображение значений функции принадлежности. Кроме того, реализована возможность загрузки и визуализации дополнительных наборов функций принадлежности для последующего анализа и сравнения.

Окно «Экспертная оценка» предоставляет интерфейс для ввода технологическим персоналом значения возможных параметры для расчета желаемой температуры в управляемом объекте.

Разработанный программный комплекс интегрируется на входе с промышленным программируемым логическим контроллером и на выходе — с последующим модулем гибридных методов управления, вычисляя ошибку управления и формируя управляющие сигналы для исполнительных устройств. Данная схема взаимодействия программных продуктов обеспечивает сквозную передачу данных от датчиков через нейро-нечеткую логику до механизмов управления в режиме реального времени, позволяет адаптировать стратегию управления при изменении технологических параметров.

Более того, модуль гибридных методов управления позволяет комбинировать адаптивный нейро-нечеткий регулятор с традиционными алгоритмами, что расширяет диапазон контроля и повышает надежность работы оборудования в целом [12-15]. Такой подход также упрощает масштабирование системы на несколько камер и интеграцию с верхнеуровневыми информационными системами предприятия для централизованного мониторинга и аналитики.

а)

Уставки Функции принадлежности Экспертная оценка

Сделать Уставки

День инкубации
1

Время инкубации
0:00

Начальная влажность
65

Начальная температура
37

Результат

Сохранить все расчеты

Сохранить температурные уставки

б)

Уставки Функции принадлежности Экспертная оценка

Просчитать температуру

День инкубации
1

Время инкубации
0:00

Влажность
65

Результат

Рассчитать

Температура в объекте **37.9**

Рис. 3. Окна разработанного программного комплекса а) «Уставки», б) «Экспертная оценка»

4. Заключение

В данной работе рассматривается задача обеспечения и поддержания оптимальных микроклиматических условий в процессе инкубации, обусловленная необходимостью непрерывного контроля температуры на всех этапах эмбрионального развития. В качестве решения предложено и реализовано специализированное программное обеспечение, в основе которого лежит нейро-нечеткий алгоритм для системы поддержки принятия решений.

Разработанная система позиционируется как экспертная поддержка технологического персонала, предоставляя рекомендации на основе данных с датчиков. Разработанное программное обеспечение имеет свидетельство о государственной регистрации программы для ЭВМ №2023687519 и успешно внедрено в производственных инкубаториях [4].

Результаты опытной эксплуатации показали, что применение нейро-нечеткого подхода способствует снижению эмбриональной смертности за счет точечной оптимизации и поддержания температурного режима в объекте управления.

Кроме того, программное обеспечение обладает модульной архитектурой, что позволяет адаптировать его под различные типы инкубационного оборудования и расширять функциональность в соответствии с требованиями конкретного производства. Интерфейс пользователя ориентирован на практикующего специалиста и предусматривает визуализацию текущих параметров, предупреждения о критических отклонениях, а также рекомендации по корректирующим действиям.

Планируется дальнейшее развитие системы в направлении интеграции и применения методов машинного обучения (так как с помощью нейро-нечеткой логики можно получить массив численных значений) для автоматической адаптации алгоритмов к специфике конкретных партий инкубуемых яиц.

Литература

1. Бессарабов Б.Ф. Инкубация яиц сельскохозяйственной птицы: Учебное пособие / Б.Ф. Бессарабов, А.А. Крыканов, А.Л. Киселев. – СПб.: Лань, 2015. – 160 с.
2. Судаков А.Н. Естественная инкубация яиц высокопродуктивных мясных кроссов кур / А.Н. Судаков, Е.А. Андрианов, А.А. Андрианов // Аграрный вестник Урала, 2020. – № 5 (196). – С. 68–79.
3. Дуванов Е.С. Особенности управления процессом инкубации / Е.С. Дуванов, Ю.И. Кудинов, Ф.Ф. Пащенко // Информатика и системы управления. – 2023. – Т. 4, № 78. – С. 18–29.
4. Дуванов Е.С. Анализ и синтез гибридных регуляторов для предиктивного управления тепловым процессом на основе нечеткой логики: автореф. дис. канд. техн. наук. — Липецк: 2024. – 24 с.
5. Duvanov E.S., Kudinov Y.I., Pashchenko F.F. Features of Egg Incubation Process Control // 2022 4th International Conference SUMMA 2022 – IEEE, 2022. – Р. 494–499.
6. Duvanov E.S., Kudinov Y.I., Pashchenko F.F., Duvanova V.S. Analysis of the technological process of egg incubation and formulation of the control problem // 2021 3rd International Conference SUMMA 2021. – 2021. – Р. 769–773.
7. Jang J.-S.R. ANFIS: Adaptive-Network-Based Fuzzy Inference System // IEEE Transactions on Systems and Cybernetics, 1993. – V. 23(3). – Р. 665–685.
8. Zadeh L.A. Fuzzy sets // Information and Control. – 1965. – № 8. – Р. 338–353.
9. Заде Л. Понятие лингвистической переменной и его применение к приятию приближенных решений // Мир. – 1976. – 165 с.
10. Кудинов Ю.И. Нечеткие модели вывода в экспертных системах // Известия РАН. Теория и системы управления. – 1997. – № 5. – С. 75–83.
11. Кудинов Ю.И. Нечеткие модели и системы управления / Ю.И. Кудинов, А.Ю. Келина, И.Ю. Кудинов, А.Ф. Пащенко, Ф.Ф. Пащенко. – М.: Ленанд, 2017. – 328.
12. Duvanov E.S., Kudinov Y.I., Pashchenko F.F., Ponomarev A.A. Analysis and synthesis of the Modified MRAC-MIT system and the MRAC-Lyapunov system // 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – IEEE, 2020. – Р. 521–526.
13. Ponomarev A.A., Kudinov Y.I., Pashchenko F.F., Duvanov E.S. Analysis and synthesis of adaptive PID Controller with MRAC-MIT system // 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – IEEE, 2020. – Р. 527–532.
14. Pashchenko F.F., Kudinov Y.I., Pashchenko A.F., Duvanov E.S. Fuzzy quadratic control of thermal object // 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA). – IEEE, 2019. – Р. 288–293.
15. Duvanov E.S., Fedyanin T.V., Pashchenko A.F. The Feasibility of Using a Predictive Neural Network Controller in the Problem of Maintaining the Water Level in a Reservoir // 2024 17th International Conference on Management of Large-Scale System Development (MLSD). – IEEE, 2024. – Р. 1–5.