

© 2026 г. А.Ф. ПАЩЕНКО, канд. техн. наук (a.paschenko@ipu.ru)  
(Институт проблем управления им. В.А. Трапезникова РАН, Москва),  
Е.С. ДУВАНОВ, канд. техн. наук (evgenyduvanov@yandex.ru)  
(Липецкий государственный технический университет)

## ГИБРИДНЫЕ РЕГУЛЯТОРЫ В ЗАДАЧАХ УПРАВЛЕНИЯ РЕАЛЬНЫМИ ОБЪЕКТАМИ

Работа продолжает исследования авторов по управлению сложными техническими объектами гибридными методами. Приводятся описание и анализ двух подходов: нейросетевых предиктивных и нечетких гибридных регуляторов. Рассматриваются вопросы адаптивной настройки, точности управления и практической реализации обоих методов. Приводятся теоретические и практические результаты применения разработанных алгоритмов в лабораторных и промышленных экспериментах.

*Ключевые слова:* гибридные методы управления, нейросетевой предиктивный регулятор, нечеткий квадратичный регулятор, управление биотехническими системами, MATLAB.

**DOI:** 10.7868/S2413977726040054

### 1. Введение

В современных системах управления все более активно внедряются гибридные подходы, объединяющие классические алгоритмы и методы искусственного интеллекта, с целью повышения адаптивности, точности и надежности. Два наиболее перспективных направления такого гибридного управления – нейросетевые предиктивные и нечеткие гибридные регуляторы. Первые позволяют предсказывать динамику сложных объектов и оптимизировать управляющие воздействия по горизонту прогнозирования, вторые – эффективно адаптироваться к неопределенностям и возмущениям на основе экспертных правил. Оба подхода демонстрируют превосходство над классическими ПИ регуляторами на практике, обеспечивая лучшие показатели качества управления в соответствующих прикладных задачах.

Искусственные нейронные сети, обладающие возможностью аппроксимации сложных нелинейных зависимостей, активно используются для построения прогностических моделей объектов управления. В традиционной схеме модель предварительно обучается на исторических данных, после чего нейросеть предсказывает будущие значения выходной величины на заданном горизонте прогнозирования. На основе этих прогнозов оптимизационный алгоритм формирует управляющее воздействие, минимизируя выбранный критерий качества. Так, в системе управления сложным механическим объектом нейросетевой предиктивный регулятор показал более высо-

кую точность и скорость реакции по сравнению с классическим ПИ регулятором [1]. Применение нейросетевой модели повышает чувствительность и быстроту реакции системы на внешние возмущения, что актуально в задачах с динамическими изменениями. Вместе с тем такие методы требуют значительных вычислительных ресурсов и объема данных для обучения сети, что осложняет их быструю практическую реализацию на промышленных объектах.

Нечеткие гибридные регуляторы представляют собой сочетание эвристических экспертных правил, формализованных в виде функций принадлежности и правил вывода, с методами оптимального управления. Гибридная структура объединяет устойчивость нечеткой логики к неопределенностям и нелинейностям с точностью и предсказуемостью классических алгоритмов, обеспечивая адаптивную корректировку управляющих уставок в реальном времени и улучшая реакцию системы на динамические возмущения.

Нечеткие гибридные регуляторы находят свое практическое применение благодаря способности учитывать сложные взаимосвязи между различными параметрами технологических процессов. Такой подход позволяет снижать колебания режимов и обеспечивать плавный и стабильный переход к требуемому значению по сравнению с традиционными ПИ регуляторами. Гибридные регуляторы в автоматических системах демонстрируют значительно более высокую точность поддержания заданных параметров, что приводит к улучшению качества продукции и снижению эксплуатационных рисков [2].

## 2. Разработка и настройка нейросетевого предиктивного регулятора

Нейросетевой предиктивный регулятор (*NNPC, Neural Network Predictive Control*) представляет собой метод предиктивного управления, адаптированный для нелинейных систем. В качестве модели используется нейронная сеть, обученная на данных экспериментов, которая на каждом такте оценки прогнозирует динамику выходных сигналов. Далее, исходя из этих прогнозов и горизонта планирования, формулируется и решается оптимизационная задача ошибки управления. Такой подход сочетает в себе преимущества предиктивного управления на основе модели и гибкость нейросетевых моделей. Целевая функция имеет вид:

$$(1) \quad J = \sum_{i=N_1}^{N_2} (y_r(t+j) - y_m(t+j))^2 + \rho \sum_{j=1}^{N_u} (u'(t+j-1) - u(t+j-2))^2,$$

где  $N_1$ ,  $N_2$  – минимальный и максимальный горизонты прогнозирования (обычно  $N_1 = 1$ ),  $N_u$  – горизонт управления,  $y_r$  – заданное эталонное значение выхода,  $y_m$  – предсказанное значение выхода сети,  $u'$  – вектор предполагаемых входных сигналов,  $\rho$  – весовой коэффициент штрафа при управлении.

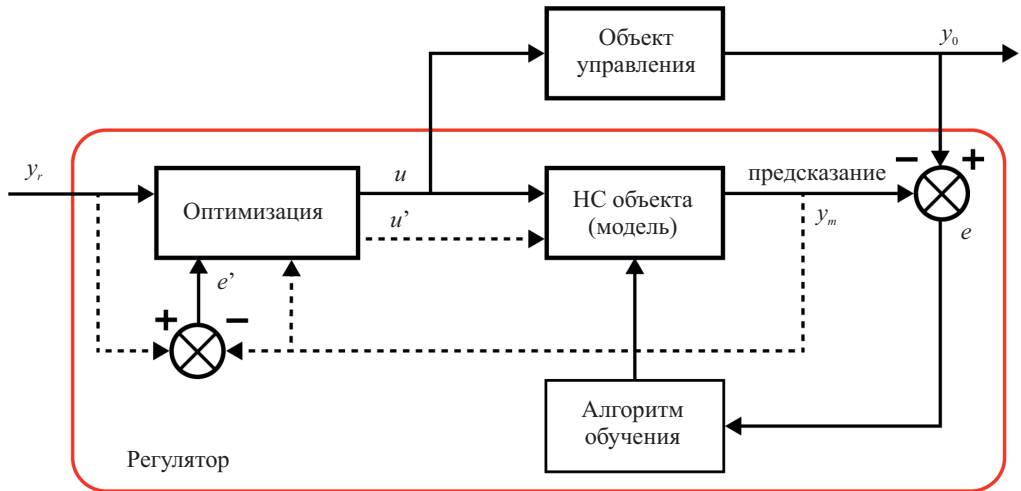


Рис. 1. Структура системы управления на основе нейросетевого предиктивного регулятора.

На рис. 1 представлена структура системы управления на основе нейросетевого предиктивного регулятора.

Сеть обучается заранее в пакетном режиме на экспериментальных данных работы реального объекта, при этом ошибка прогноза (разница между истинным выходом системы и предсказанием сети) минимизируется. Для идентификации обычно применяется двухслойная сеть *MLP* (*Multilayer Perceptron*). На вход сети подаются последние значения управлений и выходов системы (через звенья запаздывания).

Входной вектор сети включает  $u(t), u(t-1), \dots$  и  $y_p(t), y_p(t-1), \dots$  – текущие и запаздывающие значения входа и выхода системы. За этими входами следует один скрытый слой с нелинейной (сигмоидной) функцией активации. На выходе – линейная комбинация скрытого слоя, выдающая прогноз следующего выхода  $y_m(t+1)$ .

Такой подход опирается на теорему универсальной аппроксимации [3]: двухслойная нейросеть может аппроксимировать любую нелинейную функцию с достаточной точностью. После завершения обучения весовые коэффициенты нейросети сохраняются, модель интегрируется в состав регулятора для последовательного прогноза будущих откликов системы. При каждом временном шаге сеть прогоняет вход  $u'(t)$  и предыдущее состояние, выдавая значение  $y_m(t+1)$ , затем этот выход при необходимости подается обратно на вход для моделирования поведения на несколько шагов вперед в зависимости от горизонта прогнозирования.

Предположим, что сеть обучена так, что на входы (моменты времени  $t, t-1, \dots$ ) она подает одношаговый прогноз выхода  $y_m(t+1)$ . Многократное итеративное применение этой сети позволяет получить прогнозы

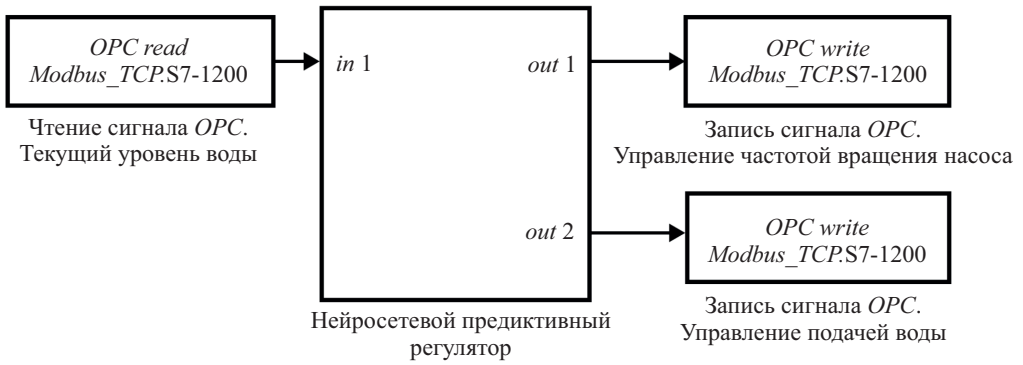


Рис. 2. Схема подключения к контроллеру с помощью OPC сервера.



Рис. 3. Блок подсистемы с нейросетевым предиктивным регулятором.

$y_m(t+2), y_m(t+3), \dots, y_m(t+N_2)$  при условии, что известны или предполагаются будущие входы  $u'(t), u'(t+1)$ .

Первая сумма в (1) накапливает квадраты ошибок слежения по горизонту  $[t+N_1, t+N_2]$ , вторая – штрафует за величину шагов управления (разницу между предполагаемыми новыми  $u'$  и последним примененным  $u$ ).

При решении задачи оптимизации критерий  $J$  минимизируется по параметрам  $u'(t), \dots, u'(t+N_u-1)$ , причем при фактическом управлении учитывается только первый элемент этого найденного вектора.

В методологии NNPC обычно фиксируют  $N=1$  (начинают учитывать ошибку сразу со следующего шага). Значение  $\rho > 0$  подбирается, чтобы сбалансировать скорость реакции регулятора и плавность управления. Модель сети представляется в виде

$$(2) \quad y_m(t+1) = \int (u(t), u(t-1), \dots, y_p(t), y_p(t-1), \dots),$$

где  $y_p(t)$  – действующий выход объекта. Например, модель описывается уравнением

$$(3) \quad y_m(t+1) = W^{(2)}\sigma\left(W^{(1)}[u(t), y_p(t)] + b^{(1)}\right) + b^{(2)},$$

где  $W^{(1)}$  и  $W^{(2)}$  – матрицы весов первого (входного) и второго (скрытого) слоев,  $b^{(1)}$  и  $b^{(2)}$  – смещения,  $\sigma$  – сигмоидная функция [4, 5].

В основе данного типа работы регулятора лежит решение задачи оптимизации на каждый такт дискретизации. Целевая функция  $J$  учитывает две составляющие: суммарную ошибку слежения по выходу за горизонт прогнозирования и штраф за изменение управляющего сигнала  $\Delta u(t)$ . При приближении к значению  $u'$  итеративный алгоритм оптимизации (метод Левенберга–Маркардта) рассчитывает градиент изменения  $J$  по параметрам управления и делает шаг к уменьшению ошибки [6].

В практической реализации на каждом шаге регулятор выполняет конечное число итераций поиска, после чего принимает текущее найденное оптимальное  $u$  и подает его на объект. Благодаря итеративной схеме и локальности аппроксимации оптимизируемой функции метод обеспечивает сходимость к локальному минимуму  $J$ , однако не гарантирует глобальный оптимум.

Возможна реализация нейросетевого предиктивного регулятора, взаимодействующего с объектом управления посредством протокола *OPC* (*Open Platform Communications*). На рис. 2 представлена схема такого подключения.

На рис. 3 представлена схема подсистемы с нейросетевым предиктивным регулятором.

### 3. Разработка и настройка нечеткого квадратичного регулятора

В 1960 г. Летовым А.М. и Калманом Р.Е. была сформулирована и успешно решена задача линейно-квадратичного управления, состоящая в нахождении оптимального управления для линейных систем с использованием квадратичных критериев. Летов А.М. разработал методику для стационарных линейных объектов, а Калман Р.Е. расширил этот подход на случай нестационарных систем [7, 8].

Линейно-квадратичный регулятор (*LQR*, *Linear Quadratic Regulator*) реализует метод оптимального управления линейными динамическими системами, основанный на минимизации квадратичной функции потерь  $e^2(t)$  с использованием квадратичных критериев. Данный метод направлен на минимизацию квадратичной функции, с учетом изменения динамики системы, входных и выходных параметров управления путем поиска оптимальной матрицы обратной связи состояния с помощью решения дифференциального уравнения Риккати [9].

Пусть линейный непрерывный объект, для которого возможно представление в виде векторно-матричного уравнения, представлен как

$$(4) \quad \dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0,$$

где  $x \in \mathbb{R}^n$  –  $n$ -мерный вектор координат объекта,  $A \in \mathbb{R}^{n \times n}$  – матрица  $[n \times n]$  параметров объекта,  $B \in \mathbb{R}^{n \times m}$  – матрица  $[n \times m]$  управления, где  $m \leq n$ ,  $u \in \mathbb{R}^m$  – вектор управления.

Уравнения состояния линейной стационарной системы можно записать в следующем виде [9]:

$$(5) \quad \begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

где  $A$  – матрица системы  $[n \times n]$ ,  $B$  – матрица входа  $[n \times m]$ ,  $C$  – матрица управления  $[k \times n]$ ,  $D$  – сквозная матрица  $[k \times m]$ .

Критерием оптимальности управления является следующее выражение:

$$(6) \quad J(u(t)) = \min \int_0^{\infty} [x^T Qx + u^T Ru] dt,$$

где матрицы  $Q$  и  $R$  – весовые матрицы для переменных состояния и входных переменных.

Целью синтеза является определение матрицы обратной связи по состоянию  $K$  с минимизацией функционала, сводящейся к решению уравнения Риккати. Функция Гамильтона имеет вид

$$(7) \quad H(t, x, u, \lambda) = -\frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu).$$

Выражение ее градиентов можно представить как

$$(8) \quad \frac{\partial H}{\partial \lambda} = Ax + Bu = \dot{x},$$

$$(9) \quad \frac{\partial H}{\partial x} = A^T \lambda - Qx = -\dot{\lambda},$$

$$(10) \quad \frac{\partial H}{\partial u} = B^T \lambda - Ru = 0.$$

Используя градиенты линейной формы  $Gz$  и квадратичной формы  $z^T Gz$ , запишем:

$$\frac{\partial}{\partial z}(Gz) = G^T, \quad \frac{\partial}{\partial z}(z^T Sz) = 2Sz,$$

где  $G$  – строка,  $S$  – симметричная матрица.

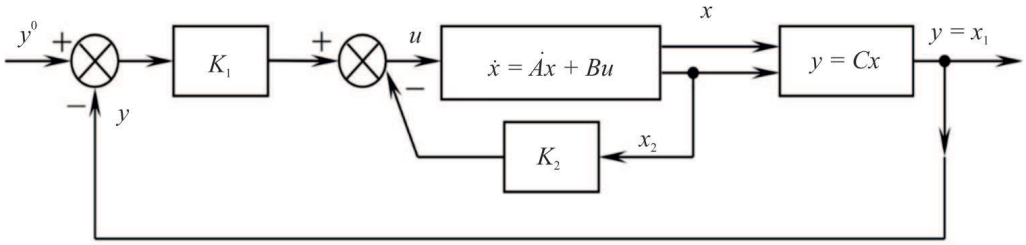


Рис. 4. Система с линейно-квадратичным алгоритмом управления.

Из уравнения (10) следует, что  $u = R^{-1}B^T\lambda$ . Исключая управление  $u(t)$  из уравнения (8), получим:

$$(11) \quad \dot{x} = Ax + BR^{-1}B^T\lambda,$$

$$(12) \quad \dot{\lambda} = Qx + A^T\lambda.$$

Для нахождения оптимального управления решим систему, полагая, что существует такая матрица  $P(t)$ , что выполняется тождество  $\lambda(t) = -P(t)x(t)$ , следовательно:

$$(13) \quad u(t) = -R^{-1}B^TP(t)x(t).$$

Далее определим необходимое и достаточное условие существования матрицы  $P(t)$ , получив матричное алгебраическое уравнение Риккати относительно матрицы  $P$ :

$$(14) \quad PA + A^TP - PBR^{-1}B^TP + Q = 0.$$

Следовательно, линейно-квадратичный алгоритм представляет собой метод, направленный на минимизацию критерия оптимальности при условии соблюдения ограничения в форме равенства, учитывающего модель объекта. Оптимальное управление может быть задано следующим образом:

$$(15) \quad u(t) = -Kx(t), \quad K = R^{-1}B^TS.$$

Отметим, что критерий качества может расширяться и для конечного горизонта или учета трекинга (с приближенными целевыми траекториями), однако в базовом варианте ставится задача стабилизации к равновесию.

К основным достоинствам классического линейно-квадратичного управления, схематичное представление которого приведено на рис. 4, относятся: гарантированная асимптотическая устойчивость замкнутой системы при наличии полного состояния, единственность решения при положительной определенности матриц и системный (структурированный) подход к синтезу управления. Данный алгоритм управления предполагает «гарантированную» устойчивость и устойчивость к малым возмущениям при адекватной моде-

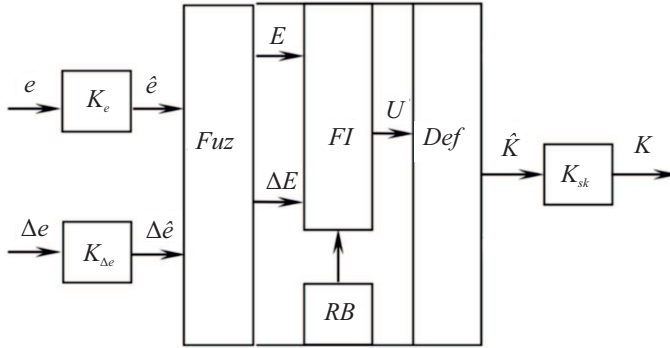


Рис. 5. Структура нечеткого квадратичного регулятора.

ли системы. Однако классический линейно-квадратичный регулятор имеет ограничения: он требует точного знания линейной модели объекта управления и допускает только квадратичную постановку задачи. В случае сильных нелинейностей или неопределенностей данный метод может оказаться неэффективным; изменение структуры или параметров системы влечет за собой пересчет регулятора.

Для расширения возможностей линейно-квадратичного регулятора используются методы на основе нечеткой логики. Нечеткий квадратичный регулятор (*FQR, Fuzzy Quadratic Regulator*) объединяет оптимальный закон с адаптивным механизмом нечеткого вывода [10]. В таких подходах структура управления дополняется блоком нечеткой логики, который на основе текущего состояния системы либо ошибки задает «поправки» к параметрам классического регулятора. Например, может вводиться «линейная функция слияния» (*Linear Fusion Function*), объединяющая выходы линейно-квадратичного регулятора и нечеткую корректировку.

Такая архитектура позволяет сочетать теоретически обоснованные свойства оптимального управления (минимизация квадратичного функционала) с гибкостью нечетких правил, обеспечивая улучшенную устойчивость, качество переходного процесса и отказоустойчивость при нелинейностях и внешних возмущениях.

В структуре нелинейной части нечеткого квадратичного регулятора имеются два входа: ошибка регулирования  $e(k)$  и скорость ее изменения  $\Delta e(k) = \Delta e(k)/\Delta t$ , а также выход  $K$  (рис. 5). С использованием нормализующих коэффициентов  $K_e, K_{\Delta e}$  фактические значения входов  $e, \Delta e$  преобразуются в нормализованные величины  $\hat{e}, \Delta \hat{e}$  в пределах интервала  $[-1, 1]$ . В процессе фаззификации осуществляется преобразование нормализованных входных величин  $\hat{e}, \Delta \hat{e}$  в нечеткие переменные  $E, \Delta E$  [1, 11].

Методом Мамдани для нечеткого вывода находится нечеткий выход  $U$  на основе нечетких входов  $E$  и  $\Delta E$ . Операция дефаззификации преобразует нечеткий выход  $U$  в нормализованное значение  $\hat{u} \in [L, -L]$  на основе мето-

да центра тяжести. Затем это нормализованное значение  $\hat{u}$  умножается на коэффициент  $K_{sk}$  для получения действительного значения  $K$ .

Настройка нечеткой составляющей квадратичного регулятора является сложной задачей, так как имеет грубую высокоуровневую настройку коэффициента, а также низкоуровневую для улучшения качества управления.

При построении нечеткой модели с двумя входами и одним выходом на этапе дефаззификации обычно применяется метод центра тяжести [1]:

$$(16) \quad \hat{y} = \frac{\int_Y yY'(y) dy}{\int_Y Y'(y) dy}.$$

На следующем этапе установления связи между входом и выходом нечеткой модели используется база правил системы нечеткого вывода. Каждое правило в этой базе состоит из условия и заключения.

Запишем переменные в непрерывной форме:  $\hat{e}(t)$ ,  $\Delta\hat{e}(t)$ ,  $u(t)$ ,  $t \in [0, T]$ . При непрерывном нечетком регуляторе правила выглядят следующим образом:

$$(17) \quad R^\theta: \text{если } \hat{e}(t) \text{ есть } E^\theta \text{ и } \Delta\hat{e}(t) \text{ есть } \Delta E^\theta, \text{ то } u(t) \text{ есть } U^\theta,$$

где  $E^\theta$ ,  $\Delta E^\theta$  и  $U^\theta$  – нечеткие множества, характеризующие ошибку  $\hat{e}(t)$ , ее скорость изменения  $\Delta\hat{e}(t)$  и управление  $u(t)$  на  $t \in [0, T]$ , принадлежащие соответствующим терм-множествам  $E \in T_e$ ,  $\Delta E \in T_{\Delta e}$ ,  $U \in T_u$  с элементами – лингвистическими значениями.

На основе анализа полученных переходных и фазовых процессов определим нечеткие правила для регулятора. Процесс формирования базы правил при нечетком квадратическом управлении включает в себя определение необходимых лингвистических переменных для входа и выхода, а также определение лингвистических терминов, описывающих значения этих переменных.

Однако стоит отметить, что если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то поддерживается текущее значение управления и  $K$  будет равно  $K\_Z$ :

$$(18) \quad R: \text{если } \hat{e} \text{ есть } E\_Z \text{ и } \Delta\hat{e} \text{ есть } DE\_Z, \text{ то } K \text{ есть } K\_Z.$$

Если ошибка  $\hat{e}$  не уточняется, то правила должны увеличивать быстродействие переходного процесса при наибольшей ошибке управления.

В процессе исследований был получен вывод, что на выходе блока нечеткой логики требуются одинаковые по своей природе, но разные по значению воздействия для каждой переменной состояния. Следовательно, на выходе блока нечеткой логики будет один коэффициент  $y$ , а масштабирующих коэффициентов несколько. Количество этих коэффициентов зависит от порядка модели. Кроме того, полученная модель нормированная, поэтому требуется определение масштабирующих коэффициентов.

В табл. 1 представлены лингвистические переменные для нечеткого квадратичного регулятора. Пояснения к сокращениям, которые часто используются для обозначения лингвистических переменных в нечеткой логике:

Таблица 1. Выбор нечетких правил для коэффициента  $K$

$\hat{e} / \Delta\hat{e}$	$NB$	$NS$	$Z$	$PS$	$PB$
$NB$	$NVB$	$NB$	$NM$	$NS$	$Z$
$NS$	$NB$	$NM$	$NS$	$Z$	$PS$
$Z$	$NM$	$NS$	$Z$	$PS$	$PM$
$PS$	$NS$	$Z$	$PS$	$PM$	$PB$
$PB$	$Z$	$PS$	$PM$	$PB$	$PVB$

$NVB$  (*Negative Very Big*) – «очень большое отрицательное»,  $NB$  (*Negative Big*) – «большое отрицательное»,  $NM$  (*Negative Medium*) – «среднее отрицательное»,  $NS$  (*Negative Small*) – «малое отрицательное»,  $Z$  (*Zero*) – «нулевое»,  $PS$  (*Positive Small*) – «малое положительное»,  $PM$  (*Positive Medium*) – «среднее положительное»,  $PB$  (*Positive Big*) – «большое положительное»,  $PVB$  (*Positive Very Big*) – «очень большое положительное» [1].

Ниже представлены нечеткие правила, основанные на табл. 1. Выбор данных нечетких правил обусловлен характером управления, ошибки и скорости ее изменения:

- $R^1$ : если  $\hat{e}$  есть  $E\_NB$  и  $\Delta\hat{e}$  есть  $DE\_NB$ , то  $K$  есть  $K\_NVB$ ,
- $R^2$ : если  $\hat{e}$  есть  $E\_NB$  и  $\Delta\hat{e}$  есть  $DE\_NS$ , то  $K$  есть  $K\_NB$ ,
- $R^3$ : если  $\hat{e}$  есть  $E\_NB$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то  $K$  есть  $K\_NM$ ,
- $R^4$ : если  $\hat{e}$  есть  $E\_NB$  и  $\Delta\hat{e}$  есть  $DE\_PS$ , то  $K$  есть  $K\_NS$ ,
- $R^5$ : если  $\hat{e}$  есть  $E\_NB$  и  $\Delta\hat{e}$  есть  $DE\_PB$ , то  $K$  есть  $K\_Z$ ,
- $R^6$ : если  $\hat{e}$  есть  $E\_NS$  и  $\Delta\hat{e}$  есть  $DE\_NB$ , то  $K$  есть  $K\_NB$ ,
- $R^7$ : если  $\hat{e}$  есть  $E\_NS$  и  $\Delta\hat{e}$  есть  $DE\_NS$ , то  $K$  есть  $K\_NM$ ,
- $R^8$ : если  $\hat{e}$  есть  $E\_NS$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то  $K$  есть  $K\_NS$ ,
- $R^9$ : если  $\hat{e}$  есть  $E\_NS$  и  $\Delta\hat{e}$  есть  $DE\_PS$ , то  $K$  есть  $K\_Z$ ,
- $R^{10}$ : если  $\hat{e}$  есть  $E\_NS$  и  $\Delta\hat{e}$  есть  $DE\_PB$ , то  $K$  есть  $K\_PS$ ,
- $R^{11}$ : если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_NB$ , то  $K$  есть  $K\_NM$ ,
- $R^{12}$ : если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_NS$ , то  $K$  есть  $K\_NS$ ,
- $R^{13}$ : если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то  $K$  есть  $K\_Z$ ,
- $R^{14}$ : если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_PS$ , то  $K$  есть  $K\_PS$ ,
- $R^{15}$ : если  $\hat{e}$  есть  $E\_Z$  и  $\Delta\hat{e}$  есть  $DE\_PB$ , то  $K$  есть  $K\_PM$ ,
- $R^{16}$ : если  $\hat{e}$  есть  $E\_PS$  и  $\Delta\hat{e}$  есть  $DE\_NB$ , то  $K$  есть  $K\_NS$ ,
- $R^{17}$ : если  $\hat{e}$  есть  $E\_PS$  и  $\Delta\hat{e}$  есть  $DE\_NS$ , то  $K$  есть  $K\_Z$ ,
- $R^{18}$ : если  $\hat{e}$  есть  $E\_PS$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то  $K$  есть  $K\_PS$ ,
- $R^{19}$ : если  $\hat{e}$  есть  $E\_PS$  и  $\Delta\hat{e}$  есть  $DE\_PS$ , то  $K$  есть  $K\_PM$ ,
- $R^{20}$ : если  $\hat{e}$  есть  $E\_PS$  и  $\Delta\hat{e}$  есть  $DE\_PB$ , то  $K$  есть  $K\_PB$ ,
- $R^{21}$ : если  $\hat{e}$  есть  $E\_PB$  и  $\Delta\hat{e}$  есть  $DE\_NB$ , то  $K$  есть  $K\_Z$ ,
- $R^{22}$ : если  $\hat{e}$  есть  $E\_PB$  и  $\Delta\hat{e}$  есть  $DE\_NS$ , то  $K$  есть  $K\_PS$ ,
- $R^{23}$ : если  $\hat{e}$  есть  $E\_PB$  и  $\Delta\hat{e}$  есть  $DE\_Z$ , то  $K$  есть  $K\_PM$ ,
- $R^{24}$ : если  $\hat{e}$  есть  $E\_PB$  и  $\Delta\hat{e}$  есть  $DE\_PS$ , то  $K$  есть  $K\_PB$ ,
- $R^{25}$ : если  $\hat{e}$  есть  $E\_PB$  и  $\Delta\hat{e}$  есть  $DE\_PB$ , то  $K$  есть  $K\_PVB$ .

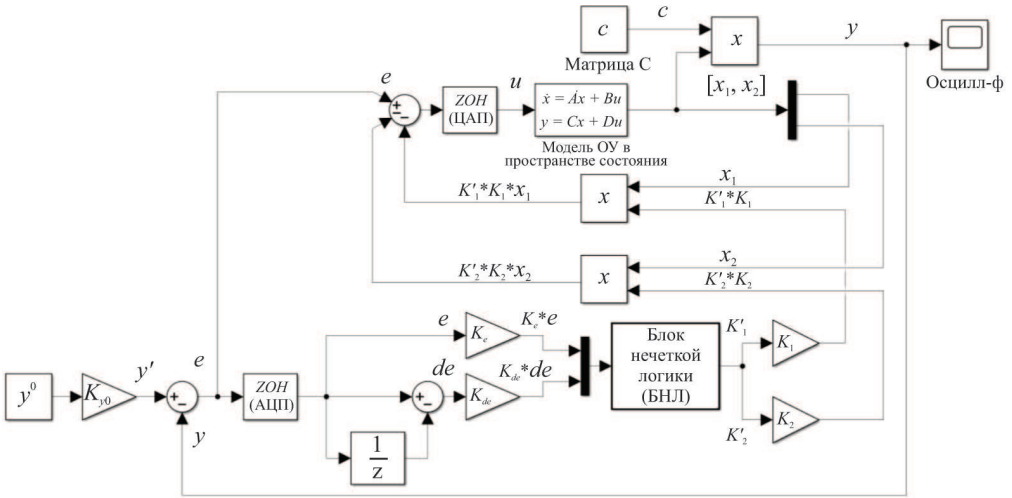


Рис. 6. Структурная схема замкнутой системы управления с нечетким квадратичным регулятором.

На рис. 6 представлена схема замкнутой системы управления с нечетким квадратичным регулятором в среде *MATLAB – Simulink*.

Задание с учетом масштабирующего коэффициента можно представить как

$$y'(t) = y^0 K_{y0}, \quad \text{ошибка управления } e(t) = y'(t) - y(t),$$

преобразование непрерывного сигнала в дискретный для работы блока нечеткой логики

$$e_{ZOH}(t) = ZOH(e(t), T),$$

дифференцирование дискретного сигнала и скорость изменения ошибки

$$de(nT) = e_{ZOH}(nT) - e_{ZOH}(nT - T).$$

Блок нечеткой логики описывается следующим образом

$$(19) \quad \begin{bmatrix} K'_1 \\ K'_2 \end{bmatrix} = \tilde{f}_{\text{БНЛ}}(K_e e_{ZOH}, K_{\Delta e} de).$$

Описание модели объекта управления в пространстве состояния

$$\dot{X}(t) = AX(t) + By(t), \quad X(t) = C \sum_t \dot{X}(t).$$

Тогда

$$X(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix},$$

следовательно,

$$U(t) = e(t) - X_1(t)K'_1(t)K_1 - X_2(t)K'_2(t)K_2.$$

Обратное преобразование дискретного сигнала в непрерывный

$$u(t) = ZOH(U(t)), \quad y(t) = CX(t)$$

преобразуется блоком «Zero-Order Hold».

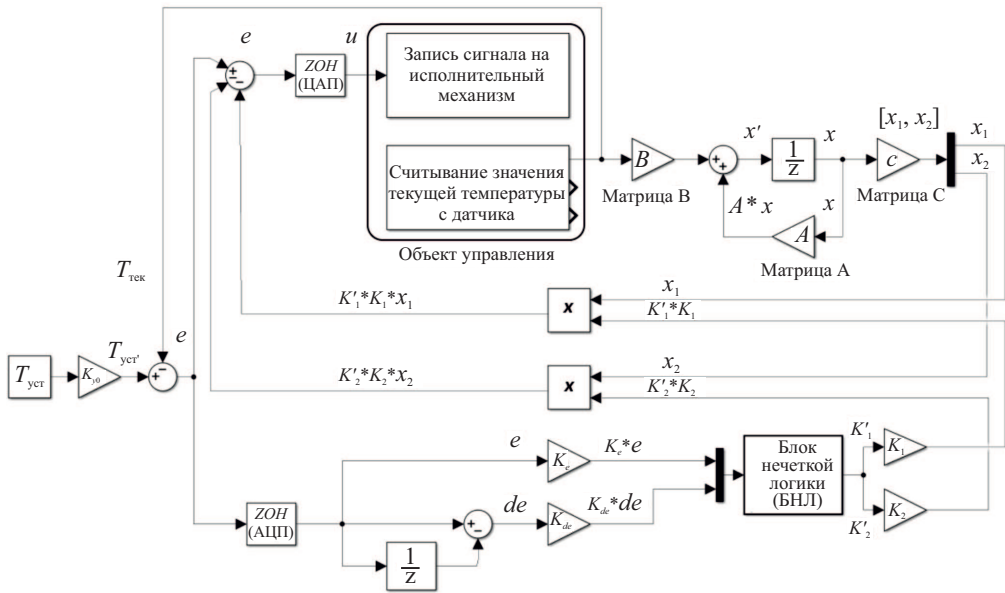


Рис. 7. Схема замкнутой системы управления с нечетким квадратичным регулятором и блоками OPC сервера.

В блоке нечеткой логики квадратичного регулятора коэффициенты  $K'_1$  и  $K'_2$  корректируются в соответствии с лингвистическими правилами. Данная процедура адаптации коэффициентов осуществляется с целью оптимизации работы регулятора, что способствует повышению эффективности и точности управления системой.

В отличие от симуляционной схемы, в схеме с OPC сервером при подключении к объекту управления необходимо добавить схему наблюдателя, который в реальном времени получает вектор состояния объекта (рис. 7).

#### 4. Реализация гибридных регуляторов на реальных объектах

В работе представлено расширенное теоретическое обоснование применения прогнозирующего нейросетевого предиктивного регулятора для задач управления объектами со сложной динамикой. Показано, что использование нейросетевых моделей позволяет эффективно аппроксимировать поведение системы на горизонте прогнозирования и, в сочетании с методами оптимизации, обеспечивает высокую точность управления. Математическая формализация критерия оптимальности, реализация стратегии горизонта прогноза и архитектура регулятора дают возможность учитывать ограничения, задержки и нелинейные свойства управляемого процесса.

На текущий момент данный тип регуляторов успешно применяется в исследовательских задачах по поддержанию уровня жидкости в гидравлическом объекте [12, 13]. При управлении использовалась программная среда *MATLAB – Simulink*, программируемый логический контроллер *Simatic S7-1200* и OPC сервер.

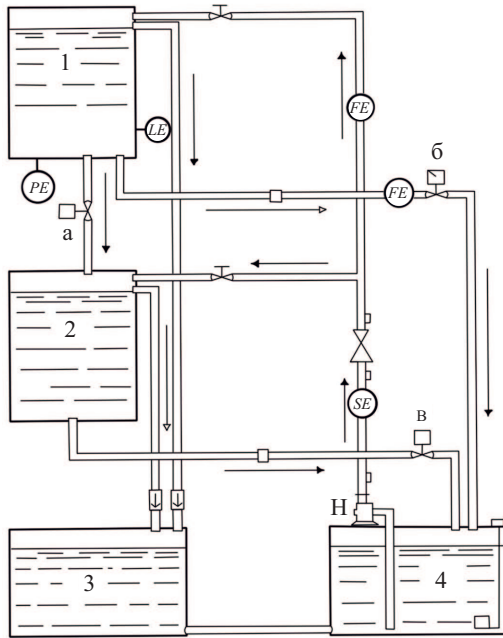


Рис. 8. Исследовательский стенд «Гидравлический объект».

Среда *MATLAB* с библиотекой *Deep Learning Toolbox* позволяет эффективно реализовать сложные системы управления на основе нейронных сетей. Блок *NN Predictive Controller* (рис. 3) выполняет функцию встроенного *NNPC* – регулятора, состоящего из входа целевого значения (*reference*), входа измеренного выхода системы (*feedback*) и выхода управляющего сигнала (*control*).

Данный блок настраивается через диалоговые окна, которые разбиты на две вкладки: *Controller Design* (параметры предиктора) и *Plant Identification* (параметры модели сети). Ключевые параметры блока следующие: горизонты предсказания и управления ( $N_1, N_2 = 1$ ), коэффициент взвешивания  $\rho$ , параметр  $\alpha$  оптимизации (определяет требуемую относительную редукцию  $J$  на каждом шаге поиска), алгоритм оптимизации и число итераций.

Вкладка *Plant Identification* задает параметры обучаемой сети-модели системы: число нейронов скрытого слоя, число задержек входных и выходных переменных, функцию обучения, а также считывания данных для обучения.

В *Simulink Plant Model* выбирается заранее подготовленная модель с передаточной функцией объекта, по которой нейросеть получит первичное представление о поведении объекта управления.

На основании серии исследований, основная задача которых заключалась в поддержании уровня жидкости в резервуаре 2 (рис. 8) на уровне 7 литров, можно сделать вывод о неустойчивости системы управления поддержанием уровня воды в резервуаре при использовании классического ПИ регулятора.

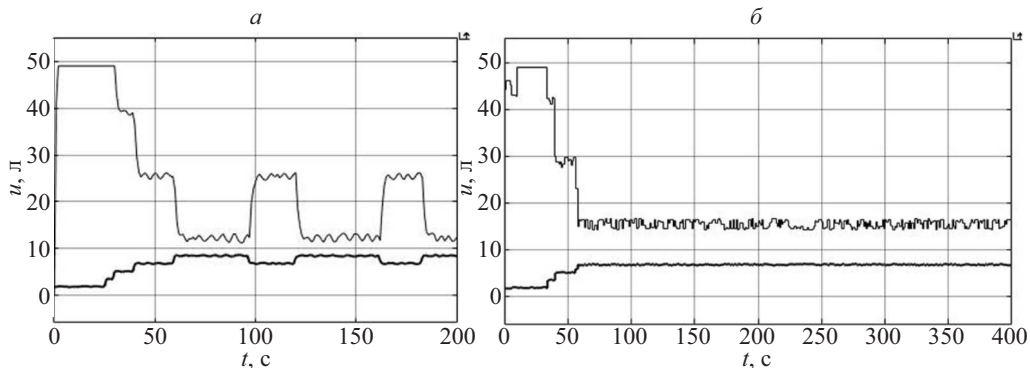


Рис. 9. Переходный процесс при использовании: *а* – классического ПИ регулятора, *б* – нейросетевого предиктивного регулятора.

Это обусловлено тем, что параметры классического ПИ регулятора не способны компенсировать минимальные накопления излишнего объема жидкости в резервуаре, приводящего к ложным реакциям на изменения (рис. 9).

Результаты экспериментальных исследований двух типов регуляторов приведены в табл. 2.

**Таблица 2.** Результаты эксперимента

Тип регулятора	Время нарастания, с	Время переходного процесса, с	Перерегулирование, л	Характер устойчивости
Нейросетевой предиктивный регулятор	54	57	0,04	Устойчива
ПИ регулятор	42	66	0,23	Неустойчива

Однако стоит отметить, что нейросетевые регуляторы могут уступать нечетким из-за своей сложности обучения и настройки, высоких вычислительных затрат, а также меньшей предсказуемости поведения в условиях помех и переходных режимов.

Нечеткие регуляторы, напротив, обеспечивают стабильность и быстродействие благодаря простым логическим правилам, минимальному времени вычислений и наглядности формирования алгоритмов управления, что делает их предпочтительными для задач с жесткими требованиями к стабильности, в частности в биотехнических системах управления, где необходимо соблюдать конкретные технологические регламенты.

В табл. 3 представлены типы и параметры функций принадлежности нечеткой составляющей квадратичного регулятора в блоке *Fuzzy Logic Toolbox* среды *MATLAB – Simulink* в задаче поддержания температурных режимов в инкубационном шкафу [9, 10, 14]. Синтез базы нечетких управляющих правил построен на методе фазовой плоскости.

**Таблица 3.** Функции принадлежности нечеткой части квадратичного регулятора

Нечеткая переменная	Функция принадлежности	Тип ФП	Параметры
E	$E\_NB$	<i>trapmf</i>	[-1,45 -1,05 -1 -0,493]
	$E\_NS$	<i>trimf</i>	[-1 -0,5 0]
	$E\_Z$	<i>trimf</i>	[-0,5 0 0,5]
	$E\_PS$	<i>trimf</i>	[0 0,5 1]
	$E\_PB$	<i>trapmf</i>	[0,506 0,999 1,16 1,34]
dE	$DE\_NB$	<i>trapmf</i>	[-1,45 -1,05 -1 -0,493]
	$DE\_NS$	<i>trimf</i>	[-1 -0,5 0]
	$DE\_Z$	<i>trimf</i>	[-0,5 0 0,5]
	$DE\_PS$	<i>trimf</i>	[0 0,5 1]
	$DE\_PB$	<i>trapmf</i>	[0,506 0,999 1,16 1,34]
K	$K\_NVB$	<i>trapmf</i>	[-1,23 -1,02 -0,987 -0,758]
	$K\_NB$	<i>trimf</i>	[-1 -0,75 -0,5]
	$K\_NM$	<i>trimf</i>	[-0,75 -0,5 -0,25]
	$K\_NS$	<i>trimf</i>	[-0,5 -0,25 0]
	$K\_Z$	<i>trimf</i>	[-0,25 0 0,25]
	$K\_PS$	<i>trimf</i>	[0,00408 0,254 0,504]
	$K\_PM$	<i>trimf</i>	[0,25 0,5 0,75]
	$K\_PB$	<i>trimf</i>	[0,5 0,75 1]
	$K\_PVB$	<i>trapmf</i>	[0,762 0,999 1,03 1,2]

Начальные значения коэффициентов  $K_1$  и  $K_2$  (рис. 6–7) вычисляются с использованием функции  $[K, S, e] = lqr(W\_ss, q, r)$ , что обеспечивает базовую настройку регулятора с учетом заданных весовых матриц  $W\_ss$ ,  $q$  и  $r$  и создает основу для последующей нечеткой коррекции согласно определенным лингвистическим правилам.

Преобразование передаточной функции объекта управления в систему в пространстве состояния осуществляется с помощью встроенной *MATLAB* функции `tf2ss/W_ss`.

На рис. 6 показана процедура формальной проверки свойств системы перед построением *LQR* регулятора с помощью функций *MATLAB* `ctrb` и `obsv`. Сначала формируется матрица управляемости  $C = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$ , с помощью функции `ctrb` определяется ее ранг. Если он равен размерности вектора состояния  $n$ , система признается полностью управляемой. Аналогично с помощью `obsv` строится матрица наблюдаемости  $Q = [C^T (CA)^T (CA^2)^T \dots (CA^{n-1})^T]^T$ , и при ее полном ранге все состояния могут быть восстановлены, что обеспечивает корректность работы наблюдателя. В случае одного входа матрица  $R$  представится в виде скалярного

**Таблица 4.** Функции принадлежности нечеткой части квадратичного регулятора

Тип управления	Перерегулирование, °С	Время переходного процесса, с
	Встроенный ПИ регулятор	
	1,9	3055
	Оптимизированный ПИ регулятор	
	1,5	3000
	Нечеткий квадратичный регулятор	
	1,0	2730

параметра  $r$ , который в оптимизирующем функционале

$$J = \min \int_0^{\infty} (x^T Q x + u^T R u) dt$$

задается эмпирически и играет ключевую роль в балансе между энергозатратами управления и быстродействием системы: при увеличении  $r$  снижается амплитуда управляющего сигнала и повышается устойчивость, но замедляются динамические реакции, тогда как уменьшение  $r$  делает регулятор более агрессивным, сокращая время перехода, но увеличивая риск перерегулирования и износа исполнительных механизмов.

После первоначального расчета коэффициентов  $K = lqr(A, B, Q, R)$  проверяют управляемость и наблюдаемость, затем в ходе серийных имитаций и экспериментальных испытаний параметр  $r$  корректируют до тех пор, пока не будут достигнуты требуемые критерии качества переходного процесса (время установления, перерегулирование) при сохранении полной управляемости и наблюдаемости. Эта итеративная оптимизация служит важным звеном, связывающим классический подход  $LQR$  и последующую нечеткую адаптацию, что позволяет добиться высокой точности и надежности управления в реальных условиях [9, 10].

Таким образом, описанный подход сочетает в себе жесткую оптимизацию через классический алгоритм  $LQR$  на этапах инициализации и гибкую адаптацию через нечеткую логику на этапе реального времени, что обеспечивает высокую точность и устойчивость управления в широком диапазоне рабочих режимов.

Активный эксперимент поддержания температурного режима в инкубационном шкафу проводился в два этапа: поиск оптимального алгоритма управления и управление полным циклом инкубации. Исследования проводились с целью изучения воздействия внешних факторов на процессы управления, осуществляемые системой, путем полного открытия двери инкубационного шкафа на промежуток времени, равный 1200 с, что имитирует краткосрочное присутствие персонала в инкубационном шкафу. Температура воздуха в инкубационном шкафу в течение часа до начала опыта – 38°С. Вентилятор принудительно включен. Результаты представлены в табл. 4.

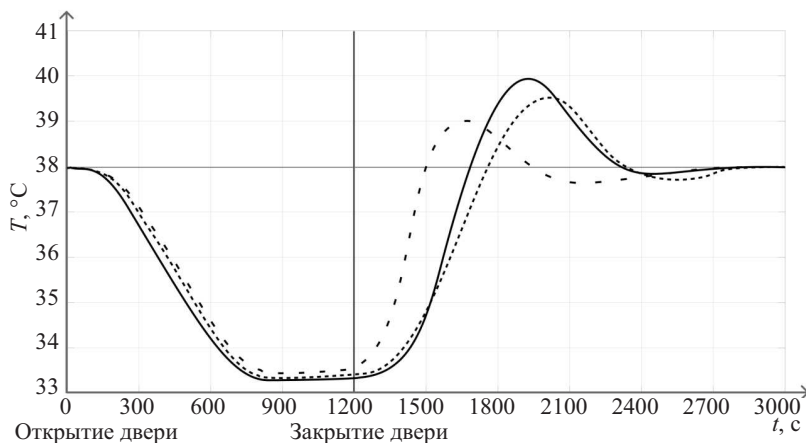


Рис. 10. График влияния внешних воздействий на объект управления в системе со встроенным (сплошная линия), оптимизированным (частая прерывистая линия), нечетким квадратичным (прерывистая линия) регуляторами.

На рис. 10 представлен график реакции внешнего воздействия на объект управления в системе.

**Таблица 5.** Основные факторы при выборе гибридного регулятора

Характеристика	Нейросетевой предиктивный регулятор	Нечеткий регулятор
Принцип работы	Использует модель объекта на основе ИНС; предсказывает будущее поведение системы и решает задачу оптимизации по целевому функционалу $J$ .	Управление определяется набором экспертных правил ЕСЛИ-ТО; входные сигналы фазсифицируются, затем выводится нечеткое управление, которое дефазсифицируется для получения конкретного значения.
Модель объекта	Идентифицируется на данных: ИНС обучается аппроксимировать динамику бака по данным эксперимента. Способна представлять сложные нелинейные зависимости.	Нет явной математической модели; динамика задается правилами (все описано «на естественном языке» через термы).
Интерпретируемость	Низкая: модель «черного ящика», трудно объяснить, почему она дает именно такое управление.	Высокая: правила понятны человеку, можно легко изменять логику управления.
Вычислительная сложность	Высокая: на этапе управления требуется решать оптимизационную задачу в реальном времени. Обучение может быть ресурсоемким.	Низкая: операции проверки правил и дефазсификации выполняются быстро, подходят для быстродействующих контроллеров.

## 5. Заключение

Выбор между нейросетевым предиктивным и нечетким регуляторами должен определяться требованиями к точности, характером динамики объекта, доступностью обучающих данных и вычислительных ресурсов, а также технологическими особенностями.

В табл. 5 представлены основные решающие факторы при выборе гибридного регулятора.

### СПИСОК ЛИТЕРАТУРЫ

1. *Пащенко Ф.Ф., Кудинов Ю.И., Пащенко А.Ф. и др.* Нечеткие модели и системы управления. М.: Ленанд, 2026. 384 с.
2. *Ignatyev V.V., Finaev V.I.* The use of hybrid regulator in design of control systems // World Applied Sciences Journal. 2013. V. 23. No. 10. P. 1291–1297.
3. *Cybenko G.V.* Approximation by Superpositions of a Sigmoidal function // Mathematics of Control Signals and Systems. 1989. V. 2. No. 4. P. 303–314.
4. *Akesson B.M., Toivonen H.T.* A neural network model predictive controller // Journal of Process Control. 2006. V. 16. No. 9. P. 937–946.
5. *Draeger A., Engell S., Ranke H.* Model predictive control using neural networks // IEEE Control Systems Magazine. 1995. V. 15. No. 5. P. 61–66.
6. *Vasickaninova A., Bakosova M., Oravec J., et al.* Neural network predictive controller design // Chemical Engineering Transactions. 2017. V. 61. P. 121–126.
7. *Лётов А.М.* Аналитическое конструирование регуляторов // АиТ. 1960. Т. 21. № 5. С. 436–441.
8. *Калман Р., Фалб П., Арбиб М.* Очерки по математической теории систем. М.: Мир, 1971. 400 с.
9. *Kudinov Y.I., Pashchenko F.F., Kelina A.Y., et al.* Analysis of control system models with conventional LQR and fuzzy LQR controller // Procedia Computer Science. 2019. V. 150. P. 737–742.
10. *Pashchenko F.F., Kudinov Y.I., Pashchenko A.F., et al.* Fuzzy quadratic control of thermal object // 2019 1st Int. Conf. on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA). IEEE, 2019. P. 288–293.
11. *Vassilyev S.N., Kudinov Yu.I., Pashchenko F.F., et al.* Intelligent Control Systems and Fuzzy Controllers II // Autom. Remote Control. 2020. V. 80. P. 345–357.
12. *Duwanov E.S., Fedyanin T.V., Pashchenko A.F.* The Feasibility of Using a Predictive Neural Network Controller // Proc. 17th Int. Conf. on Management of Large-Scale System Development (MLSD). IEEE, 2024. P. 1–5.
13. *Duwanov E.S., Batishchev R.V., Pashchenko A.F., et al.* Applying of Predictive Neural Network Controllers // Proc. 4th Int. Conf. on Technology Enhanced Learning in Higher Education (TELE). IEEE, 2024. P. 422–427.
14. *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.

*Статья представлена к публикации членом редколлегии А.А. Галяевым.*

Поступила в редакцию 08.07.2025

После доработки 15.10.2025

Принята к публикации 20.11.2025