

УДК 004.89 + 681.51  
ББК 32.813

## **О ПРИМЕНЕНИИ НЕЙРОСЕТЕВОГО ОПТИМИЗАТОРА ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА ДЛЯ УПРАВЛЕНИЯ НАГРЕВАТЕЛЬНЫМИ ПЕЧАМИ В РАЗЛИЧНЫХ РЕЖИМАХ РАБОТЫ**

**Еременко Ю. И.<sup>1</sup>, Полеценко Д. А.<sup>2</sup>, Глуценко А. И.<sup>3</sup>**  
*(Старооскольский технологический институт  
им. А.А. Угарова (филиал) ФГАОУ ВПО НИТУ «МИСиС»,  
Старый Оскол)*

*Рассмотрена проблема автоматической настройки коэффициентов ПИ-регулятора в реальном масштабе времени при управлении нагревательными объектами. Такую настройку предлагается осуществлять с помощью нейросетевого оптимизатора, включающего две нейронных сети, отвечающих за подбор коэффициентов  $K_p$ ,  $K_I$  для процессов нагрева или охлаждения печи соответственно. Приведена структура предлагаемого оптимизатора. Показана применимость подобного оптимизатора к объектам управления с различными постоянными времени. Приведена реализация системы управления для муфельной электронагревательной печи, работающей в различных режимах. Результаты проведенных экспериментов позволяют сделать вывод о том, что использование оптимизатора с подобной структурой позволяет экономить до 23% времени и 19% электроэнергии на выполнение графика заданий по сравнению с обычным ПИ-регулятором.*

---

<sup>1</sup> Юрий Иванович Еременко, доктор технических наук, профессор (erem49@mail.ru).

<sup>2</sup> Дмитрий Александрович Полеценко, кандидат технических наук, доцент (po-dima@yandex.ru).

<sup>3</sup> Антон Игоревич Глуценко, кандидат технических наук (г. Старый Оскол, м-н Макаренко, д. 42, тел. (4725) 45-12-17, stronduitt@mail.ru).

Ключевые слова: нейронная сеть, адаптивное управление, ПИ-регулятор, нейросетевой оптимизатор.

## **1. Введение**

В настоящее время, несмотря на большое число имеющихся методов и законов управления, подавляющее большинство контуров управления технологическими процессами в условиях реальных производств построены на основе ПИД-регуляторов. Данный факт связан с тем, что в целом данный закон управления является достаточно простым для понимания и реализации, а также надежным в работе. Однако, обладая такими преимуществами, он остается линейным, в то время как большинство реальных объектов управления (ОУ) обладают нелинейными характеристиками. В этой связи применение ПИД-регуляторов для объектов, являющихся в значительной степени нелинейными, сопряжено с определенными трудностями. Примером таких объектов являются нагревательные ОУ – печи нагрева литой заготовки, дуговые печи и пр. Изменение задания, графика заданий, степени загрузки печи при постоянных коэффициентах ПИД-регулятора зачастую приводит к снижению качества регулирования и к росту энергозатрат на ведение технологических процессов, поскольку для подобных ОУ принудительное охлаждение невозможно, ввиду чего наличие даже небольшого перерегулирования требует длительного времени на охлаждение [16].

Теоретически для каждого состояния ОУ необходимо подбирать свои значения коэффициентов регулятора. Однако в реалиях производства чаще всего используют всего один набор таких коэффициентов для всех режимов и не производят их перенастройку ввиду трудоемкости данного процесса.

Для решения указанной проблемы необходимо построение адаптивной системы управления технологическим процессом [10]. Для создания таких систем разработано достаточно большое число методов. Среди них можно выделить группу классических методов, таких как:

1) метод Циглера–Никольса [20], в большинстве случаев позволяющий получать коэффициенты, значения которых достаточно далеки от оптимальных;

2) методы, основанные на построении модели ОУ с помощью специального испытательного сигнала (ступенчатого или гармонического): частотный метод А.Г. Александрова [8], метод, заложенный в основу работы адаптивных ПИД-регуляторов фирмы Siemens [16] (на подобном же принципе основаны адаптивные регуляторы фирм Omron, Control Station и др.) и пр.

Эти методы предполагают полный отказ от используемых в настоящее время в регуляторе коэффициентов и подбор новых. Моменты времени, когда необходимо использовать метод адаптации, определяются человеком, как и амплитуда испытательного сигнала. Кроме того, метод, описанный в [8] сталкивается с определенными сложностями при идентификации ОУ с постоянной времени более 800 с, а для метода, описанного в [16], результаты идентификации ОУ даже в сходных условиях могут весьма существенно отличаться. В целом же построение модели реального нелинейного объекта управления является весьма нетривиальной и трудноосуществимой в реалиях производства задач.

Кроме того, для решения описанной проблемы могут быть использованы интеллектуальные методы [1, 2, 6, 11, 12, 17]. Среди них можно выделить подходы, в рамках которых настройка ПИД-регулятора осуществляется с помощью нечеткой логики [7, 19] и экспертных систем [9]. Это примеры безмодельных методов настройки ПИД-регулятора. В частности, в таких экспертных системах в базе правил указано на сколько процентов следует изменить амплитудное значение того или иного коэффициента регулятора в зависимости от текущей ситуации, описанной в условии конкретного правила. Проблемой подобных подходов является отсутствие механизма обучения, поскольку первоначально степень нелинейности ОУ достаточно тяжело оценить, в связи с чем в базу правил (в частности, в их следствия) ЭС или нечеткой системы необходимо будет вносить изменения уже в процессе работы.

Для решения задач управления среди интеллектуальных методов исследователи выделяют, в частности, нейронные сети (НС), поскольку НС обладают нелинейными свойствами и способностью к обучению, что придает адаптивные свойства нейросетевым системам управления. В частности, Сигеру Омату в [15] предложил схему управления на основе автонастройки коэффициентов ПИД-регулятора  $K_P$ ,  $K_I$ ,  $K_D$  с помощью НС (рис. 1, где  $e$  – рассогласование).



Рис. 1. Схема управления с нейросетевым оптимизатором параметров регулятора

Такая схема позволяет учитывать нелинейные свойства объекта, не внося значительных изменений в существующую схему управления. Необходимо лишь подключить НС к существующей системе управления, что существенно снижает стоимость внедрения такой интеллектуальной надстройки.

Однако реализация идей С. Омату в том виде, в котором они были изложены в его трудах, не всегда позволяет получать заявленные результаты. Это во многом связано с тем, что НС не обладает сведениями о специфике того объекта, которым управляет. Такая специфика может быть задана, например, набором правил, определяющим ситуации, в которых обучение НС целесообразно.

В рамках данной работы предлагается скомбинировать метод применения НС для коррекции коэффициентов с базой правил ситуаций, когда необходимо проводить такую настройку. В качестве следствий в правилах будут выступать скорости обучения нейронов НС.

В рамках данного исследования рассмотренная выше схема была реализована в среде Matlab. Предложенный метод построения нейросетевого оптимизатора был проверен на моделях лабораторных муфельных электронагревательных печей СНОЛ-1,6.2,5.1/11-И4 и СНОЛ 40/1200 и на самой печи СНОЛ-1,6.2,5.1/11-И4. Было проведено сравнение работы систем управления с нейросетевым оптимизатором, включающим в себя две нейронные сети, и обычным ПИ-регулятором для различных режимов работы ОУ по загрузке и графику заданий.

## 2. Постановка задачи

Объектом исследований является схема управления, представленная на рис. 1. В качестве ОУ рассматривается нагревательная печь. Такой ОУ в большинстве случаев описывается аperiodическим звеном второго порядка со звеном задержки или двумя аperiodическими звеньями первого порядка со звеном задержки (1). Первое из них является моделью самой печи, а второе – моделью термопары, с помощью которой ведется измерение температуры внутреннего пространства печи:

$$(1) \quad W(s) = \frac{K}{T_1 s + 1} \cdot \frac{1}{T_2 s + 1} \cdot e^{-\tau \cdot s},$$

где  $K$  – коэффициент усиления печи;  $T_1$  – постоянная времени печи, с;  $T_2$  – постоянная времени термопары, с;  $\tau$  – время запаздывания, с.

Таким образом, процессы, происходящие внутри печи, могут быть представлены с помощью дифференциального уравнения

$$(2) \quad T_1 T_2 y''(t) + (T_1 + T_2) y'(t) + y(t) = K \cdot u(t - \tau),$$

где  $u$  – управляющее воздействие;  $t$  – текущий момент времени, с;  $y(t)$  – выход ОУ в текущий момент времени, °С.

В большинстве случаев для управления нагревательными объектами используется ПИ-регулятор, поскольку Д-канал существенно подвержен влиянию помех. Тогда уравнение (2) можно представить в виде

$$(3) \quad T_1 T_2 y''(t) + (T_1 + T_2) y'(t) + y(t) = K \cdot (K_P e(t - \tau) + K_I \int_0^{t-\tau} e(p) dp),$$

где  $e$  – рассогласование между уставкой по температуре и реальной температурой в печи,  $^{\circ}\text{C}$ ;  $K_P$ ,  $K_I$  – коэффициенты ПИ-регулятора.

Такое уравнение отражает одно конкретное состояние ОУ, поскольку  $T_1$  и  $K$  здесь являются константами. Для реального ОУ они являются переменными. Постоянная времени печи может изменяться в зависимости от количества материала, загруженного в эту печь. Коэффициент усиления уменьшается по мере увеличения температуры в печи, а кроме того, зависит от качества энергоносителя, подаваемого на печь.

Задачей контура управления, изображенного на рис. 1, является отслеживание графика уставок, представляющий собой набор ступенчатых изменений задания, таким образом, чтобы, с одной стороны, переходные процессы протекали за минимальное количество времени. Это позволит интенсифицировать производство и сократить потери энергоносителя. С другой стороны, качество получаемых переходных процессов по переуправлению, колебательности, статической ошибке должно находиться в пределах допуска.

Соответственно, для поддержания времени переходного процесса близким к минимальному в условиях изменения параметров  $T_1$ ,  $K$  необходимо оперативно в процессе работы настраивать коэффициенты регулятора  $K_P$ ,  $K_I$ .

В данном случае предполагается, что ОУ уже функционирует под управлением ПИ-регулятора и его коэффициенты изначально подобраны для одного из состояний объекта. Задачей является подстройка (а не полное переопределение) коэффициентов ПИ-регулятора в соответствии с изменениями режима работы печи для поддержания времени переходных процессов для всех режимов близким к минимальному.

Для этого предлагается применить нейросетевой оптимизатор параметров ПИ-регулятора. Он не предполагает построение модели ОУ. Вместо этого производится оценка ситуации на ОУ (в частности, текущей скорости переходного процесса) с помо-

щью базы правил с дальнейшими выводами о необходимости обучения НС.

### **3. Структура нейросетевого оптимизатора**

Предлагаемый обобщенный алгоритм работы нейросетевого оптимизатора параметров ПИИ-регулятора приведен на рис. 2.

При работе с нагревательными объектами необходимо вести управление двумя различными по своей природе процессами: нагреванием (когда новая уставка больше предыдущей) и охлаждением (когда новая уставка меньше предыдущей). В ходе проведенных исследований [2] было выявлено, что довольно часто для оптимальности ведения и того, и другого процесса требуется два набора коэффициентов регулятора, а не один, поскольку нагревательная печь неспособна вести принудительное охлаждение своего внутреннего пространства. Например, значение коэффициента  $K_I$  может быть оптимальным для процесса нагрева, а при охлаждении из-за медленной скорости процесса такое значение  $K_I$  может привести к снижению качественных показателей переходной характеристики из-за насыщения  $i$ -канала.

В связи с этим предлагается реализовывать нейросетевой оптимизатор в виде двух идентичных нейронных сетей, каждая из которых отвечает за управление коэффициентами регулятора для своего типа процесса (нагрева и остывания).

Переключение между НС предлагается производить в момент очередной смены задания. При этом определяется тип переходного процесса (нагрев или охлаждение) и производится выбор соответствующей сети. Обе сети начинают работу с формирования на своем выходе одинаковых коэффициентов регулятора, используемых на ОУ в текущий момент. Метод достижения этой цели описан в [3].

До начала каждого из экспериментов создаются две необученные НС. Они имеют одинаковую трехслойную структуру, так как в соответствии с [13] любая непрерывная функция может быть аппроксимирована с необходимой точностью трехслойной НС. Количество и вид входов НС были определены

экспериментальным путем в ходе исследований. Полученные результаты совпали с результатами, приведенными С. Омату в [15]. Во входном слое расположено 5 нейронов: текущее задание по температуре ( $x_1$ ); сигнал по выходу объекта управления, задержанный на 1 секунду ( $x_2$ ), на  $Z$  секунд ( $x_3$ ) и на  $2 \cdot Z$  секунд ( $x_4$ ); задержанный на 1 секунду сигнал управления, формируемый ПИ-регулятором ( $x_5$ ). То есть  $\vec{X} (x_1, x_2, x_3, x_4, x_5)$  – это вектор входных сигналов в НС.

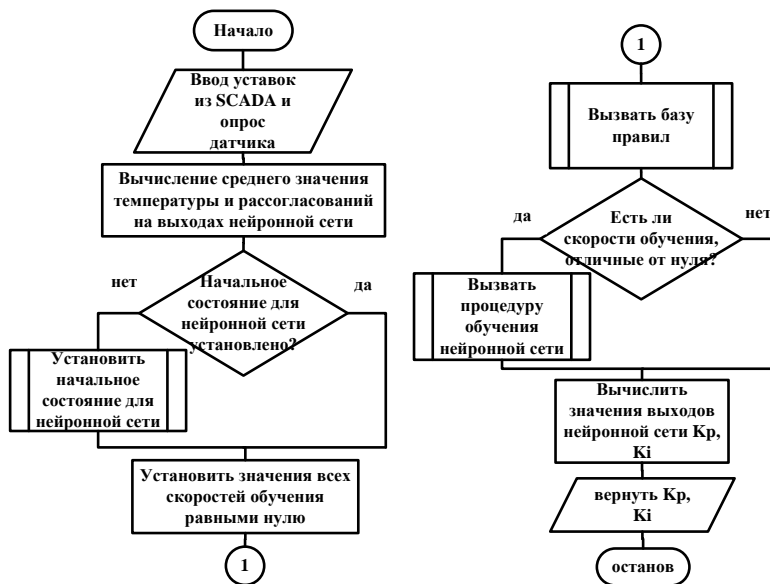


Рис. 2. Разработанный обобщенный алгоритм работы нейросетевого оптимизатора

Дискретность по задержкам  $Z$  предлагается определять следующим образом. Обращение к самой НС происходит раз в  $Z$  секунд. То есть НС оценивает состояние ОУ через равные промежутки времени, равные  $Z$  секунд. В результате проведенных экспериментов [2–5] было выявлено, что для корректной работы нейросетевой надстройки необходимо, чтобы в течение конкретного переходного процесса в период от начала процесса



до входа контролируемой величины в 10% окрестность новой уставки таких оценок было проведено не менее пятнадцати. То есть за время переходного процесса необходимо получить минимум 15 измерений температуры через равные промежутки времени  $Z$  секунд. Поскольку ОУ уже используется, то для текущих настроек регулятора время переходного процесса  $t_{пп}$  уже известно. Тогда для определения значения  $Z$  необходимо выбрать наиболее короткий по времени переходный процесс среди имеющихся в текущем графике заданий и разделить его время на 15. Например, в случае, рассмотренном в п. 7–8, время переходного процесса составляло 300 с, тогда  $Z = 300 / 15 = 20$  с.

В дальнейшем планируется ввести в оптимизатор функцию расчета времени  $Z$ , которая будет автоматически перерасчитывать это значение (в сторону уменьшения), если было получено недостаточное количество измерений температуры в течение текущего переходного процесса.

Проведенные исследования [5] показали, что 15 нейронов в скрытом слое является оптимальным числом. В скрытом слое была использована функция активации «гиперболический тангенс», в выходном слое – линейная функция. С. Омату в своей работе [15] также использовал НС с 15 нейронами в скрытом слое для управления различными по своей динамике ОУ.

В выходном слое в общем случае используется три нейрона (отвечают за коэффициенты ПИД-регулятора  $K_P$ ,  $K_I$ ,  $K_D$ ). В данном случае речь идет о ПИ-регуляторе, поэтому используется лишь два нейрона выходного слоя, а выход третьего всегда равен нулю.

Выбранная структура нейронной сети приведена на рис. 3.

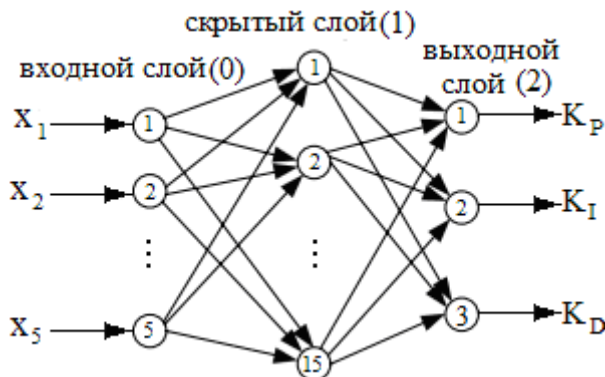


Рис. 3. Структура нейронной сети, применяемой в нейросетевом оптимизаторе параметров ПИД-регулятора

Математическая модель такой сети:

$$s_j^{(1)} = \sum_{i=1}^5 \omega_{ji}^{(1)} \cdot x_i + b_j^{(1)},$$

$$O_j^{(1)} = f^{(1)}(s_j^{(1)}) \quad (j = \overline{1, 15}),$$

$$s_k^{(2)} = \sum_{j=1}^{15} \omega_{kj}^{(2)} \cdot O_j^{(1)} + b_k^{(2)},$$

$$O_k^{(2)} = f^{(2)}(s_k^{(2)}) \quad (k = \overline{1, 3}),$$

где  $\omega_{ji}^{(1)}$  – весовой коэффициент связи между  $j$ -м нейроном скрытого слоя и  $i$ -м нейроном входного слоя;  $\omega_{kj}^{(2)}$  – весовой коэффициент связи между  $k$ -м нейроном выходного слоя и  $j$ -м нейроном скрытого слоя;  $x_i$  –  $i$ -й входной сигнал НС;  $b_j^{(1)}$  – линейное смещение  $j$ -го нейрона скрытого слоя;  $b_k^{(2)}$  – линейное смещение  $k$ -го нейрона выходного слоя;  $O_j^{(1)}$  – выходной сигнал  $j$ -го нейрона скрытого слоя;  $O_k^{(2)}$  – выходной сигнал  $k$ -го нейрона выходного слоя;  $s_j^{(1)}$  – взвешенная сумма для  $j$ -го нейрона скрытого слоя;  $s_k^{(2)}$  – взвешенная сумма для  $k$ -го нейрона выходного слоя;  $f^{(q)}$  – функция активации слоя номер  $q$ . В частности,  $f^{(1)}$  – гиперболический тангенс;  $f^{(2)}$  – линейная функция активации;  $O_1^{(2)}$  – это  $K_P$ ,  $O_2^{(2)}$  – это  $K_I$ ,  $O_3^{(2)}$  – это  $K_D$ .

Обе НС нейросетевого оптимизатора предлагается оперативно обучать с помощью алгоритма обратного распространения ошибки. Математическая модель метода в адаптированном для рассматриваемой НС виде представлена в виде (5)–(10). Общей целью обучения является минимизация рассогласования между выходом ОУ и заданием по температуре:

$$(5) \quad E(t) = \frac{1}{2} (r(t) - y(t))^2 \rightarrow \min ,$$

$$(6) \quad e_2(t) = r(t) - y(t), \quad e_1(t) = e_2(t) - e_2(t-1), \\ e_3 = e_2(t) - 2e_2(t-1) + e_2(t-2),$$

$$(7) \quad \delta_k^{(2)} = e_k \frac{dO_k^{(2)}}{ds_k^{(2)}}, \quad k = \overline{1, 3},$$

$$(8) \quad \delta_j^{(1)} = \sum_{k=1}^3 \delta_k^{(2)} \omega_{kj}^{(2)} \frac{dO_j^{(1)}}{ds_j^{(1)}}, \quad j = \overline{1, 15},$$

$$\Delta \omega_{kj}^{(2)}(t) = \eta_k^{(2)} \delta_k^{(2)} O_j^{(1)} + \alpha \Delta \omega_{kj}^{(2)}(t-1) + \beta \Delta \omega_{kj}^{(2)}(t-2),$$

$$(9) \quad \Delta b_k^{(2)}(t) = \eta_k^{(2)} \delta_k^{(2)} + \alpha \Delta b_k^{(2)}(t-1) + \beta \Delta b_k^{(2)}(t-2),$$

$$\Delta \omega_{ji}^{(1)}(t) = \eta_j^{(1)} \delta_j^{(1)} O_i^{(0)} + \alpha \Delta \omega_{ji}^{(1)}(t-1) + \beta \Delta \omega_{ji}^{(1)}(t-2),$$

$$\Delta b_j^{(1)}(t) = \eta_j^{(1)} \delta_j^{(1)} + \alpha \Delta b_j^{(1)}(t-1) + \beta \Delta b_j^{(1)}(t-2),$$

$$\omega_{kj}^{(2)}(t+1) = \omega_{kj}^{(2)}(t) + \Delta \omega_{kj}^{(2)}(t),$$

$$(10) \quad b_k^{(2)}(t+1) = b_k^{(2)}(t) + \Delta b_k^{(2)}(t),$$

$$\omega_{ji}^{(1)}(t+1) = \omega_{ji}^{(1)}(t) + \Delta \omega_{ji}^{(1)}(t),$$

$$b_j^{(1)}(t+1) = b_j^{(1)}(t) + \Delta b_j^{(1)}(t),$$

где  $E(t)$  – целевая функция процесса оперативного обучения;  $r(t)$  – это задание по температуре;  $y(t)$  – выход ОУ;  $\eta^{(1)}$  – скорость обучения для нейронов скрытого слоя;  $\eta_k^{(2)}$  – скорость обучения для  $k$ -го нейрона выходного слоя;  $\alpha$  и  $\beta$  – коэффициенты инерции, применяемые для ускорения сходимости;  $\delta_j^{(1)}$  – совокупная ошибка  $j$ -го нейрона скрытого слоя;  $\delta_k^{(2)}$  – совокупная ошибка  $k$ -го нейрона выходного слоя;  $e_k$  – ошибка  $k$ -го нейрона выходно-

го слоя;  $dO_j^{(q)}/ds_j^{(q)}$  – производная функции активации слоя  $q$ . В соответствии с [15] параметр  $\alpha$  был принят равным 0,8, а  $\beta$  – равным  $(-0,15)$ . Скорость обучения для всех нейронов скрытого слоя  $\eta^{(1)}$  является одинаковой и не меняется в течение работы системы. Численное значение данной величины также заимствовано из работы [15] и составляет  $10^{-4}$ .

В отличие от [15], в данной работе предлагается значения  $\eta_k^{(2)}$  сделать различными для нейронов выходного слоя НС и изменять их в течение работы системы. Пусть  $\eta_1^{(2)}$  – скорость коррекции весовых коэффициентов и смещения, отвечающих за формирование выхода НС  $O_1^{(2)}$ , т.е.  $K_P$ . Соответственно  $\eta_2^{(2)}$  отвечает за  $K_I$  и  $\eta_3^{(2)}$  – за  $K_D$ . Необходимость такого подхода вызвана, во-первых, тем фактом, что порядок значений коэффициентов регулятора обычно различается (например,  $K_P = 2 \cdot 10^{-1}$ , а  $K_I = 1,5 \cdot 10^{-4}$ ), поэтому и скорость их настройки должна быть пропорциональна этому значению. Во-вторых, нет необходимости изменять коэффициенты регулятора постоянно, а в некоторые моменты коррекции требует только один из них.

Поэтому в начале каждой итерации работы нейросетевого оптимизатора все три указанные скорости приравниваются к нулю, а затем происходит вызов предлагаемой базы правил. Условия правил описывают ситуации, когда необходимо использовать и обучать НС, а следствия – необходимые значения  $\eta_1^{(2)}$ ,  $\eta_2^{(2)}$ ,  $\eta_3^{(2)}$  (см. п.4). Если какое-либо из правил срабатывает, то происходит обучение НС.

Данная база правил разработана по результатам многочисленных экспериментов. Рассмотрим ее структуру подробнее.

#### **4. База правил для определения значения скорости обучения отдельных нейронов**

Дальнейшее описание базы правил будет приведено для ПИ-регулятора, наиболее широко применяемого в системах управления нагревательными технологическими объектами, поэтому  $\eta_3^{(2)}$  всегда равна нулю, так же как и  $O_3^{(2)}$ .

Рассмотрим подробнее сами правила выбора скорости обучения, сформированные на основе анализа работы нейросетевого оптимизатора при управлении нагревательными объектами.

#### *4.1. ОПРЕДЕЛЕНИЕ МОМЕНТОВ, КОГДА СЛЕДУЕТ ВЕСТИ ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ*

Во-первых, обучать НС следует, только если сигнал управления, формируемый ПИ-регулятором, находится в допустимых пределах (например,  $[0; 100]$ ). В противном случае коэффициенты регулятора приобретут излишне высокое (или низкое) значение, что приведет к насыщению интегрального канала управления. Более того, если сигнал управления с ПИ-регулятора вышел за указанные пределы, следует выдавать значение коэффициента  $K_I = 0$ .

Во-вторых, после очередной смены задания обучение стоит вести, только если рассогласование по контролируемой величине стало менее 95% от разницы между текущим заданием и предшествующим, поскольку до этого момента, ввиду наличия запаздывания в ОУ, оценка скорости нарастания сигнала температуры будет близка к нулю, что вызовет срабатывание ряда правил, модифицирующих коэффициенты регулятора.

#### *4.2. ПРАВИЛА ДЛЯ ВЫБОРА СКОРОСТИ ОБУЧЕНИЯ ПРИ УПРАВЛЕНИИ ПРОЦЕССОМ НАГРЕВА ПЕЧИ*

Если идет процесс нагрева и рассогласование по температуре находится в пределах от 10% до 80% разницы между текущим заданием и предыдущим, и:

1) скорость нарастания сигнала температуры менее амплитуды шума  $N$  в установившемся режиме, определяемой при первичном запуске нейросетевого оптимизатора, за  $Z$  секунд, то переходный процесс достиг своего пика раньше необходимого и следует вызывать блок правил по обучению нейрона выходного слоя, ответственного за формирование коэффициента  $K_P$ , выход которого в данном случае следует увеличивать. Скорость обучения такого нейрона  $\eta_1^{(2)}$  должна быть на 3 порядка меньше значения  $K_P$ . Для остальных нейронов выходного слоя при этом используются скорости обучения  $\eta_2^{(2)}$ ,  $\eta_3^{(2)}$ , равные нулю;

2) скорость нарастания сигнала температуры более допустимой по технологической инструкции, то следует вызывать блок правил по обучению нейрона выходного слоя, ответственного за формирование коэффициента  $K_P$ , выход которого в данном случае следует уменьшить. Скорость обучения такого нейрона  $\eta_1^{(2)}$  должна быть на 4 порядка меньше значения  $K_P$ . Для остальных нейронов выходного слоя при этом используются скорости обучения  $\eta_2^{(2)}$ ,  $\eta_3^{(2)}$ , равные нулю.

Если идет процесс нагревания и предыдущий блок условий не был вызван, то после уменьшения рассогласования ниже 20% от разницы между текущим заданием и предыдущим сработает другой блок условий. Алгоритм его работы следующий:

1. Если в момент очередной смены задания рассогласование между заданием и температурой в печи составляет менее 3% от разницы между текущим заданием и предыдущим и в течение предшествующего переходного процесса система управления была устойчива (см. п. 5), то текущее задание и сигнал управления сохраняются в специальный массив. Такие воздействия принимаются как эталонные для соответствующих заданий. В течение любого переходного процесса при первом пересечении задания фиксируется значение сигнала управления в  $i$ -канале. Если это значение меньше эталонного, то принимается решение об увеличении  $K_I$ , а если больше – то об уменьшении.

2. Если принято решение об увеличении  $K_I$  и перерегулирование при этом оказалось более 5% от разницы между текущим заданием и предыдущим, то принимается решение об уменьшении  $K_P$ . Для нейрона, формирующего  $K_P$ , скорость обучения  $\eta_1^{(2)}$  должна быть на 4 порядка меньше значения коэффициента  $K_P$ .

3. Обучение нейрона, формирующего  $K_I$ , для увеличения его выходного сигнала начинается при соблюдении следующих условий: переходный процесс уже достиг своего первого максимума после пересечения задания, скорость изменения температуры за последние 2 такта работы схемы (1 такт – это  $Z$  секунд) менее  $N$  градусов, а текущее рассогласование больше 5% от разницы между текущим заданием и предыдущим. Обучение этого нейрона в сторону уменьшения выходного сигнала начи-

нается при соблюдении следующих условий: скорость изменения температуры за последние 2 такта работы схемы более  $N$  градусов, а текущее рассогласование больше 5% от разницы между текущим заданием и предыдущим. Скорость обучения  $\eta_2^{(2)}$  в этих случаях должна быть на 7 порядков меньше значения  $K_I$ .

#### 4.3. ПРАВИЛА ДЛЯ ВЫБОРА СКОРОСТИ ОБУЧЕНИЯ ПРИ УПРАВЛЕНИИ ПРОЦЕССОМ ОСТЫВАНИЯ ПЕЧИ

В случае если происходит остывание печи, предложен следующий алгоритм проверки условий:

1) если скорость убывания температуры менее  $N$  градусов за два такта работы схемы, а ошибка более 10% разницы между текущим заданием и предыдущим, и график температуры в течение текущего переходного процесса еще не пересекал задание, то принимается решение об увеличении значения  $K_I$ . Скорость обучения  $\eta_2^{(2)}$  в этих случаях должна быть на 7 порядков меньше значения  $K_I$ ;

2) если график температуры пересекал задание, температура в текущий момент больше задания, а ошибка составляет более 5% разницы между текущим заданием и предыдущим, то принимается решение об уменьшении  $K_P$ . Скорость обучения такого нейрона  $\eta_1^{(2)}$  должна быть на 4 порядка меньше значения  $K_P$ ;

3) Такое же решение принимается, если график температуры пересек задание один раз, а ошибка в текущий момент более 5% разницы между текущим заданием и предыдущим. Скорость обучения такого нейрона  $\eta_1^{(2)}$  должна быть на 4 порядка меньше значения  $K_P$ .

После проверки всех блоков условий вызывается алгоритм обратного распространения ошибки, корректирующий значения весов и смещений в соответствии с выбранными с помощью правил скоростями обучения для каждого нейрона.

## **5. Об определении устойчивости предложенной системы управления**

В соответствии с [18, 19], из-за того, что коэффициенты регулятора являются функциями времени и изменяются даже в течение конкретного переходного процесса, оценка устойчивости подобной системы классическими математическими методами является весьма затруднительной. Возможно лишь оценивать стабильность конкретного переходного процесса и, в случае обнаружения неустойчивости, возвращать регулятору коэффициенты, которые использовались в рамках последнего устойчивого переходного процесса.

Существует ряд критериев для оценки устойчивости конкретного переходного процесса. В рамках данной работы использовались подходы, описанные Андерсоном [9] и Неслером [14]. Система считалась неустойчивой если:

1) график температуры более трех раз пересекал задание и каждый раз амплитуда колебаний возрастала;

ИЛИ

2) разница между значением температуры и задания в любой момент времени после первого пересечения задания графиком температуры достигла более 30% разницы между текущим заданием и предыдущим.

В случае выполнения одного из этих пунктов адаптация должна быть отключена, а регулятору должны быть возвращены первоначальные коэффициенты, которые использовались в течение последнего устойчивого переходного процесса.

## **6. Результаты моделирования работы нейросетевого оптимизатора**

Для нейросетевого оптимизатора проведена оценка границ его применимости при управлении ОУ типа (2). Результаты приведены в [4].

Ниже продемонстрирована работа предлагаемого подхода для нескольких моделей нагревательных печей с постоянными времени различных порядков.



В пакете Matlab была реализована схема управления (рис. 4) объектом, представленным блоком *Subsystem* с ПИ-регулятором (*Subsystem*) с нейросетевым оптимизатором (*S-function neuC\_PI*). Здесь  $U_i$ ,  $U_p$ ,  $U_{pi}$  – сигналы управления на выходе  $i$ -,  $p$ -каналов и всего ПИ-регулятора соответственно;  $e$  – текущее рассогласование. Блок *Saturation* использован для ограничения получаемого с ПИ-регулятора сигнала управления до диапазона  $[0; 100]$ .  $U$  – управляющее воздействие, подаваемое на ОУ. Блок *Subsystem* реализует ОУ типа (2), на выход которого наложена помеха амплитудой  $N$ , равной  $1^{\circ}\text{C}$ .

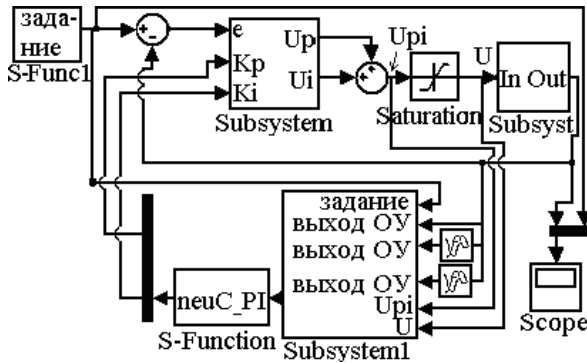


Рис. 4. Реализация в Matlab схемы управления с нейросетевым оптимизатором параметров ПИ-регулятора

Блок *Subsystem1*, формирующий вектор входов нейросетевого оптимизатора, имеет шесть входов. Первые пять из них являются входами НС, описанной в п. 3. Шестой вход необходим для корректной работы базы правил, приведенной в п. 4.

Для моделирования в качестве объекта использовались математические модели незагруженных и загруженных металлическими заготовками муфельных электропечей СНОЛ-1,6,2,5.1/11-И4 и СНОЛ 40/1200, полученные путем идентификации. Загрузка составляла ориентировочно 20–30% от рабочего пространства каждой печи.

Параметры для моделей печи СНОЛ-1,6.2,5.1/11-И4:

- 1) пустая:  $K = 21$ ,  $T_1 = 1636$  с,  $T_2 = 69,4$  с,  $\tau = 64$  с;
- 2) загруженная:  $K = 21$ ,  $T_1 = 2683$  с,  $T_2 = 35$  с,  $\tau = 90$  с.

Параметры для моделей печи СНОЛ 40/1200:

- 1) пустая:  $K = 34,2$ ,  $T_1 = 9000$  с,  $T_2 = 100$  с,  $\tau = 12$  с;
- 2) загруженная:  $K = 34,2$ ,  $T_1 = 22000$  с,  $T_2 = 100$  с,  $\tau = 20$  с.

Реализованы они были в пакете Matlab в виде  $S$ -функций при помощи метода структурного моделирования. Для изменения состояния ОУ непосредственно в процессе моделирования производилась замена набора параметров модели (2): модель незагруженной печи (12 смен задания)  $\rightarrow$  модель загруженной печи (12 смен задания)  $\rightarrow$  модель незагруженной печи (12 смен задания).

Исходя из технологического регламента для печей нагрева металлургических предприятий, был сформирован график температурного режима нагрева литой заготовки в масштабе 1:2 как по времени, так и по температуре (рис. 4, блок  $S$ -Func1 «Задание»). Смена заданий по температуре производилась путем повторения последовательности значений:  $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  и т.д. Для смены уставки температура должна удерживаться в течение 30 минут в окрестности задания, равной 5% разности текущего задания и предыдущего.

Для рассматриваемых моделей незагруженных печей были подобраны значения коэффициентов ПИ-регулятора, обеспечивающие не более 5% перерегулирования для рассматриваемого графика заданий. Для модели печи СНОЛ-1,6.2,5.1/11-И4:  $K_P = 0,2422$ ,  $K_I = 1,504 \cdot 10^{-4}$ , для модели печи СНОЛ 40/1200:  $K_P = 1,2$ ,  $K_I = 1,2 \cdot 10^{-4}$ . Нейросетевой оптимизатор начинал работу с формирования этих значений коэффициентов на своих выходах.

В соответствии с изложенным в п. 3, параметр  $Z$  для модели печи СНОЛ-1,6.2,5.1/11-И4 был рассчитан как  $Z = 300 / 15 = 20$  с. Для печи СНОЛ 40/1200  $Z = 150 / 15 = 10$  с.

Результаты моделирования для схем управления с нейросетевым оптимизатором и обычным ПИ-регулятором с использо-

ванием моделей печи СНОЛ-1,6.2,5.1/11-И4 приведены на рис. 5–6.

Перерегулирование для переходных процессов при использовании модели загруженной печи для обычного ПИ-регулятора составило 8% для процессов нагрева и 2% для процессов охлаждения, для системы с нейросетевым оптимизатором – 2% для нагрева и 2% – для охлаждения. Общее время опыта для системы с нейросетевым оптимизатором составило 27,88 часа, а для обычного ПИ-регулятора – 30,86 часа (экономия по времени на выполнение графика задания – 9,7%).

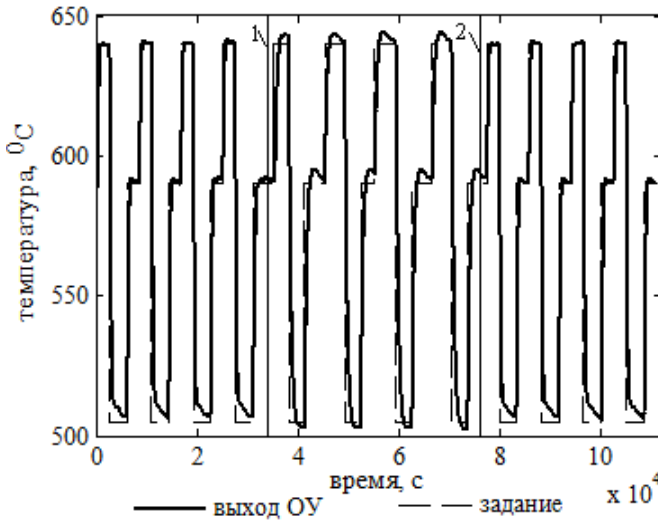


Рис. 5. Результаты для моделей печи СНОЛ-1,6.2,5.1/11-И4 для обычного ПИ-регулятора. 1 – момент начала использования модели загруженной печи; 2 – момент возврата к модели пустой печи

Результаты моделирования для схем управления с нейросетевым оптимизатором и обычным ПИ-регулятором с использованием моделей печи СНОЛ 40/1200 приведены на рис. 7–8.

Перерегулирование для переходных процессов при использовании модели загруженной печи для обычного ПИ-регулятора

для процессов нагрева составило 10%, а для системы с нейросетевым оптимизатором – 4,4%. Общее время опыта для системы с нейросетевым оптимизатором составило 36,09 часа, а для обычного ПИ-регулятора – 42,16 часа (экономия по времени на выполнение графика задания – 14,3%).

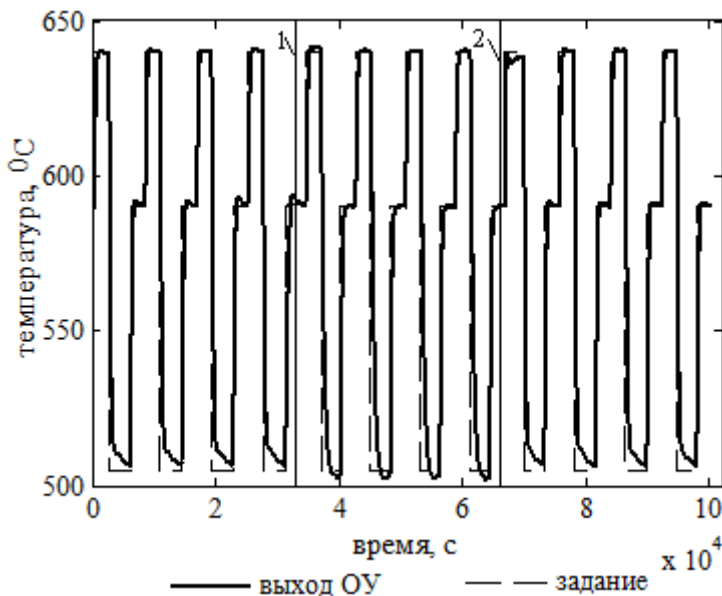


Рис. 6. Результаты для моделей печи СНОЛ-1,6.2,5.1/11-И4 для нейросетевого оптимизатора. 1 – момент начала использования модели загруженной печи; 2 – момент возврата к модели пустой печи

В целом проведенные исследования позволяют сделать вывод о том, что предлагаемый метод применим к нагревательным печам с различной динамикой и размерами.

В процессе исследований были выявлены следующие ограничения на применение метода: 1) постоянная времени печи должна быть более 500 секунд; 2) задание должно изменяться ступенчато; 3) соотношение транспортного запаздывания к постоянной времени печи должно быть менее 0,1 (аналогичное

ограничение накладывается, например, на применение адаптивного регулятора фирмы Siemens [16]); 4) минимальная величина ступенчатого изменения задания, для которой нейросетевой оптимизатор сохраняет работоспособность, прямо пропорционально зависит от  $N$ .

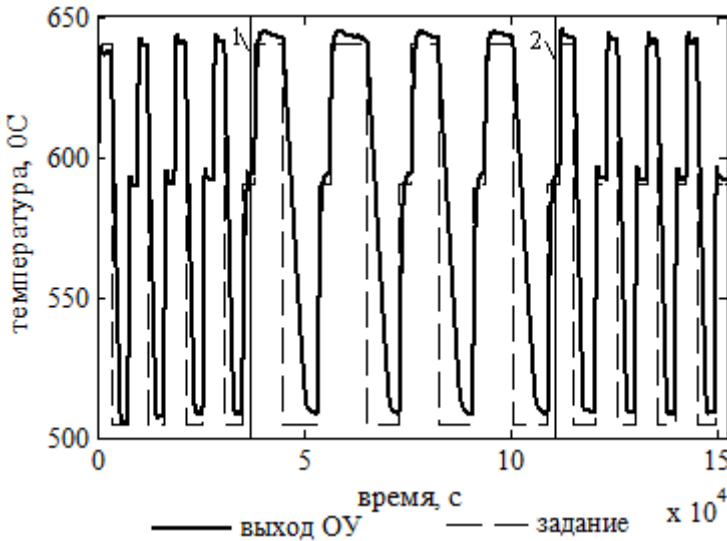


Рис. 7. Результаты для моделей печи СНОЛ 40/1200 для обычного ПИ-регулятора. 1 – момент начала использования модели загруженной печи; 2 – момент возврата к модели пустой печи

В соответствии с условиями ряда правил из базы правил, приведенной в п. 4, в течение конкретного переходного процесса текущее рассогласование сравнивается с определенным процентом от разницы текущей уставки и предыдущей. Минимальная такая величина, записанная в условии ряда правил, составляет 5%. Соответственно 5% от величины смены уставки должны быть больше амплитуды помех  $N$ , иначе оценки скоростей изменения контролируемой величины станут невозможны. Тогда минимально допустимая величина ступенчатого изменения задания должна составлять  $100\% \cdot N^{\circ}C / 5\% = 20 \cdot N^{\circ}C$ .

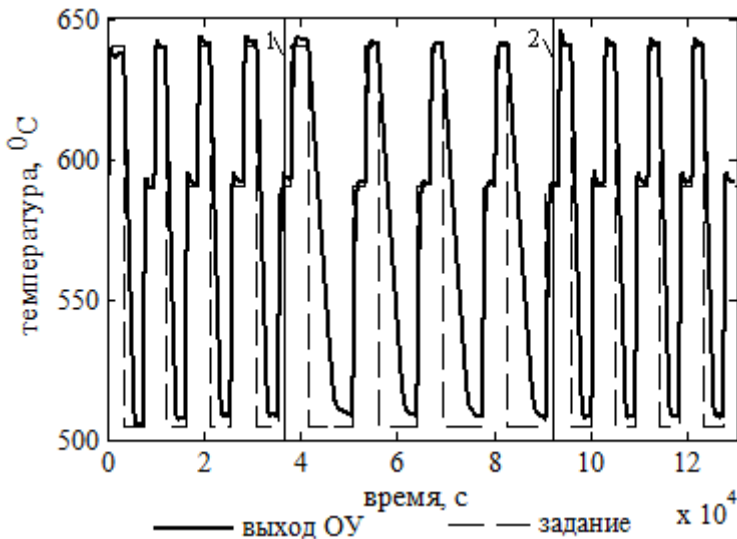


Рис. 8. Результаты для моделей печи СНОЛ 40/1200 для нейросетевого оптимизатора. 1 – момент начала использования модели загруженной печи; 2 – момент возврата к модели пустой печи

## 7. Исследование работы нейросетевого оптимизатора на реальном объекте управления

### 7.1. ОПИСАНИЕ КОНТУРА УПРАВЛЕНИЯ ПЕЧЬЮ СНОЛ-1,6,2,5.1/11-И4

В качестве ОУ для натуральных экспериментов была выбрана муфельная электронагревательная печь типа СНОЛ-1,6,2,5.1/11-И4. Реализованная система управления муфельной электропечью, функциональная схема которой представлена на рис. 9, состояла из трех уровней управления.

На нижнем уровне находились термопара, измеряющая температуру рабочей зоны печи, и реле, посредством симисторов включающее нагревательный элемент.

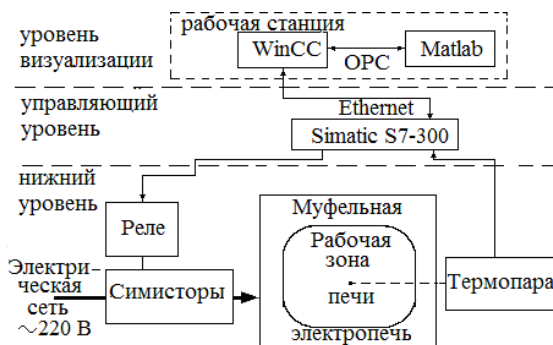


Рис. 9. Функциональная схема системы управления печью

Управляющий уровень (Simatic S7-300) осуществлял непосредственное управление муфельной электропечью. Контроллер получал данные от термопары через термопреобразователь, производил их обработку и отправлял на рабочую станцию, от которой он получал значение управляющего воздействия. В контроллере производилась широтно-импульсная модуляция управляющего воздействия и его выдача на исполнительный механизм включения нагревательного элемента.

На уровне визуализации находилась рабочая станция с системой визуализации WinCC и пакетом Matlab, в котором реализован нейросетевой оптимизатор параметров ПИ-регулятора. WinCC и Matlab взаимодействовали между собой через OPC-интерфейс посредством тегов.

Схема управления, реализованная в пакете Matlab, приведена на рис. 10. WinCC передавал данные о температуре (*temper*) и уставку (*zadanie*) по температуре в Matlab. Запущенная в Matlab НС формировала значение управляющего воздействия (*upravl*), которое через WinCC поступало в контроллер. *Group1*, *group2* – это OPC блоки соответственно для записи и чтения значений в тегах WinCC.

В Matlab реализован нейросетевой оптимизатор (см. блок *S-function neuc\_PI*, рис. 10) параметров ПИ-регулятора (*PI*). Здесь  $U_i$ ,  $U_p$ ,  $U_{pi}$  – сигналы управления на выходе *i*-, *p*-каналов и всего ПИ-регулятора соответственно. Блок *satur* использован для ограничения сигнала управления с ПИ-регулятора до диапа-

зона  $[0; 100]$  единиц.  $U$  – управляющее воздействие, подаваемое на ОУ.

Блок нейросетевого оптимизатора *S-function* имеет шесть входов. Первые пять из них являются входами НС, описанной в п. 3. Шестой вход необходим для корректной работы базы правил, приведенной в п. 4.

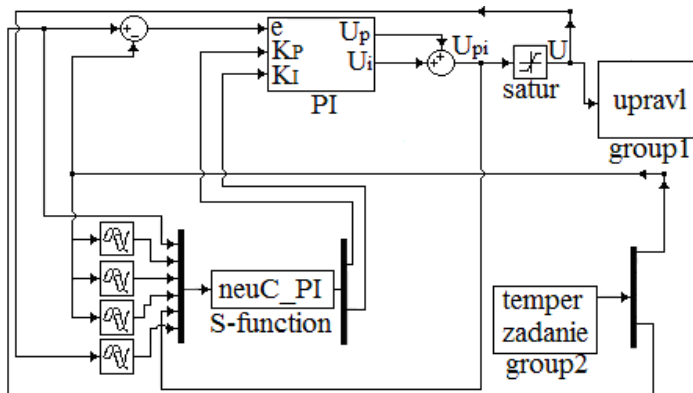


Рис. 10. Схема управления печью, реализованная в Matlab.

Для первого опыта график смены заданий выглядел следующим образом:  $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  и т.д. Для смены уставки температура должна удерживаться в окрестности задания, равной 5% разницы между текущим заданием и предыдущим, в течение 30 минут. Для второго опыта был использован следующий график заданий:  $590^{\circ}\text{C} \rightarrow 620^{\circ}\text{C} \rightarrow 570^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$ , сохраняющий прежнюю базовую температуру, но при этом амплитуда ступенчатых смен заданий выбрана меньше. Для третьего опыта был выбран график заданий с измененной базовой температурой и амплитудой смен задания:  $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$  и т.д.

Для рассматриваемой пустой печи были подобраны коэффициенты ПИ-регулятора. Для переходов  $505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  и  $590^{\circ}\text{C} \rightarrow 620^{\circ}\text{C}$ :  $K_p = 0,6$ ,  $K_i = 6,976 \cdot 10^{-4}$ . Для перехода  $240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$ :  $K_p = 0,37$ ,  $K_i = 2,66 \cdot 10^{-4}$ .



## **7.2. РЕЗУЛЬТАТЫ НАТУРНЫХ ЭКСПЕРИМЕНТОВ**

В рамках исследования были проведены три пары экспериментов (каждая пара – для одного из графиков изменения задания). Каждая пара состояла из двух одинаковых по условиям экспериментов для двух систем управления – для системы с нейросетевым оптимизатором параметров ПИ-регулятора и для обычного ПИ-регулятора.

Эксперимент заключался в том, что исследовалась работа системы в следующих условиях: с начального момента и до момента, когда задание было изменено 12 раз, муфельная печь была пустой, далее в нее загружалась литая заготовка объемом примерно 30% от рабочего пространства печи, и задание изменялось еще 12 раз, а затем печь разгружалась и проводилось еще 12 переходных процессов. Каждый переходный процесс считался завершенным, если после входа в окрестность задания, равную 5% разницы между текущим заданием и предыдущим, проходило 30 минут. Затем происходила смена задания. Для этого был использован счетчик времени. Если температура в печи выходила за пределы окрестности, то счетчик времени обнулялся. Нейросетевой оптимизатор всегда начинал эксперимент с известными значениями коэффициентов ПИ-регулятора, указанными ранее как оптимальные для пустой печи. Обычный ПИ-регулятор всегда использовал только их.

Результаты экспериментов, полученные с использованием нейросетевого оптимизатора параметров ПИ-регулятора и обычного ПИ-регулятора, для графика смены заданий  $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  приведены на рис. 11 и рис. 12.

До момента загрузки печи болванкой обе системы ведут себя практически одинаково. В период работы загруженной печи система с нейросетевым оптимизатором позволила добиться 4% перерегулирования для процессов нагрева и 2,2% – для охладжений, в то время как для обычного ПИ-регулятора для процессов нагрева этот показатель составил 12%, для охлаждения – 4,5%. После разгрузки печи ПИ-регулятор с первого переходного процесса начинает работать как до загрузки печи. Системе с нейросетевым оптимизатором требуется некоторое время, что-

бы вернуть параметры регулятора к исходным. Затем системы начинают работать практически идентично. Общее время опыта для системы с нейросетевым оптимизатором составило 29,08 часа при расходе электроэнергии 15,11 кВт\*ч, а для обычного ПИ-регулятора – 37,7 часа при расходе электроэнергии 17,56 кВт\*ч. Таким образом, экономия по времени на выполнение графика задания при использовании нейросетевого оптимизатора составила 23%, а электроэнергии – 14%.

Результаты экспериментов, полученные с использованием нейросетевого оптимизатора параметров ПИ-регулятора и обычного ПИ-регулятора, для графика смены заданий  $590^{\circ}\text{C} \rightarrow 620^{\circ}\text{C} \rightarrow 570^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  приведены на рис. 13 и рис. 14. Перерегулирование для переходных процессов при загруженной печи для обычного ПИ-регулятора составило 11% для процессов нагрева и 12% для процессов охлаждения, для системы с нейросетевым оптимизатором – 6% для нагрева и 4% – для охлаждения. Общее время опыта для системы с нейросетевым оптимизатором составило 24,1 часа при расходе электроэнергии 12,9 кВт\*ч, а для обычного ПИ-регулятора – 31,36 часа при расходе электроэнергии 16,6 кВт\*ч (экономия по времени на выполнение графика задания – 23%, по расходу электроэнергии – 22%).

Результаты экспериментов, полученные с использованием нейросетевого оптимизатора параметров ПИ-регулятора и обычного ПИ-регулятора, для графика смены заданий  $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$  приведены на рис. 15 и рис. 16. Перерегулирование для переходных процессов нагрева в течение периода использования загруженной печи для обычного ПИ-регулятора составило 10% для нагрева и 4% для охлаждения, для системы с нейросетевым оптимизатором эти показатели составили 6% для нагревов и 2,2% – для охлаждений. Общее время опыта для системы с нейросетевым оптимизатором составило 30,2 часа, а расход электроэнергии – 6,46 кВт\*ч, а для обычного ПИ-регулятора – 40,08 часа и 8,2 кВт\*ч. Таким образом, можно сделать вывод об экономии по времени в 24% и по расходу электроэнергии в 21% на выполнение графика задания при использовании нейросетевого оптимизатора.

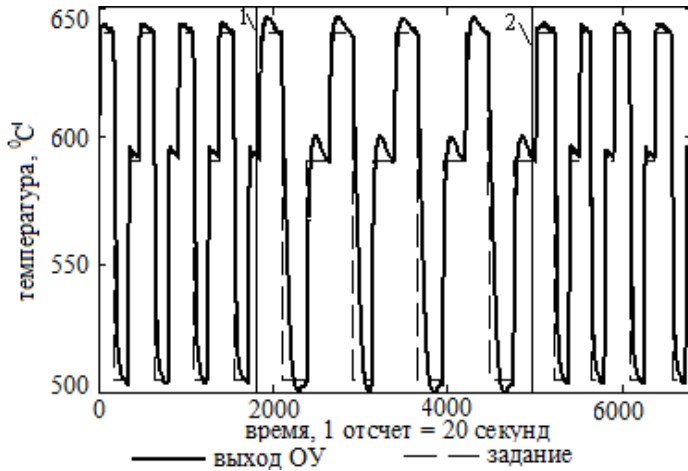


Рис. 11. Результаты для графика изменения задания  $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  для обычного ПИ-регулятора. 1 – момент загрузки печи; 2 – момент разгрузки печи

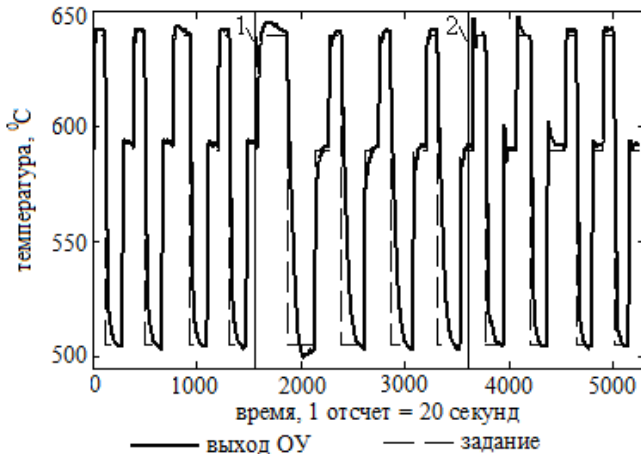


Рис. 12. Результаты для графика изменения задания  $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  для нейросетевого оптимизатора. 1 – момент загрузки печи; 2 – момент разгрузки печи

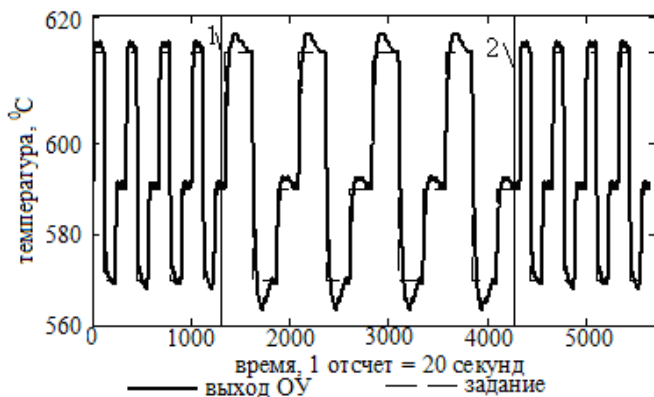


Рис. 13. Результаты для графика изменения задания  $590^{\circ}\text{C} \rightarrow 620^{\circ}\text{C} \rightarrow 570^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  для обычного ПИ-регулятора. 1 – момент загрузки печи; 2 – момент разгрузки печи

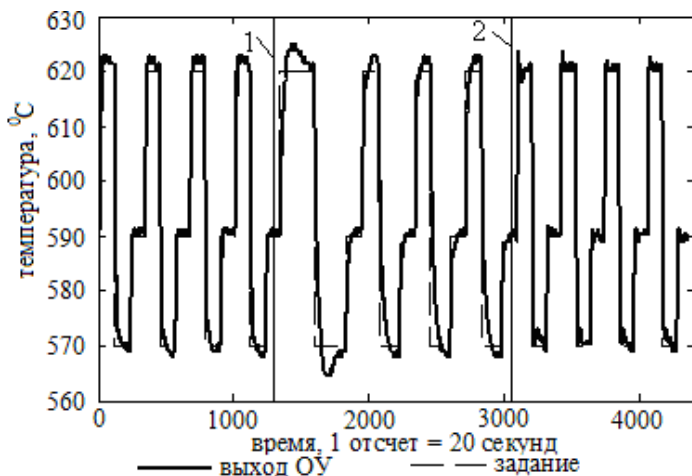


Рис. 14. Результаты для графика изменения задания  $590^{\circ}\text{C} \rightarrow 620^{\circ}\text{C} \rightarrow 570^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$  для нейросетевого оптимизатора. 1 – момент загрузки печи; 2 – момент разгрузки печи

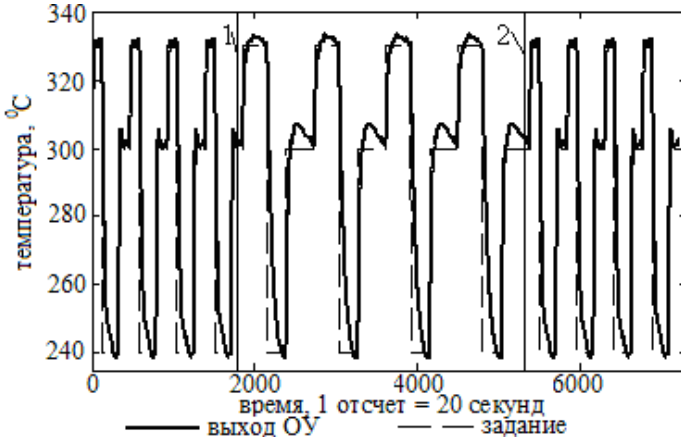


Рис. 15. Результаты для графика изменения задания  $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$  для системы управления с обычным ПИ-регулятором, 1 – загрузка печи; 2 – разгрузка печи

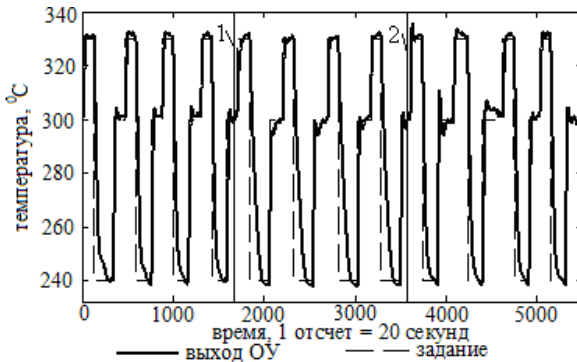


Рис. 16. Результаты для графика изменения задания  $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$  для системы управления с нейросетевым оптимизатором. 1 – загрузка печи; 2 – разгрузка печи

## 8. Заключение

Предложенный подход к построению нейросетевого оптимизатора параметров ПИ-регулятора проверен как на моделях нагревательных объектов с различной динамикой, так и на реальной лабораторной печи.

Полученные результаты позволяют сделать вывод о том, что использование нейросетевого оптимизатора параметров ПИ-регулятора для учета нелинейных свойств ОУ позволило сократить перерегулирование и время, требуемое на выполнение графика заданий, а также существенно сократить расход электроэнергии. Таким образом, в условиях реального производства на данном агрегате может быть интенсифицировано производство продукции и сокращено удельное энергопотребление.

В течение всех опытов система управления оставалась устойчивой с точки зрения выбранных критериев.

В дальнейшем предполагается исследовать возможность использования нейросетевого оптимизатора для управления ОУ, обладающими более высокой динамикой.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (гранты 13-08-00532-а, 15-07-0609215) и фонда Президента РФ (грант №14.У30.15.4865-МК)

### Литература

1. ВАСИЛЬЕВ В.И., ИДРИСОВ И.И. *Алгоритмы проектирования и анализа устойчивости интеллектуальной системы управления* // Вестник Уфимского государственного авиационного технического университета. – 2008. – Т. 11, №1. – С. 34–42.
2. ЕРЕМЕНКО Ю.И., ПОЛЕЩЕНКО Д.А., ГЛУЩЕНКО А.И. *Исследование применимости нейросетевого оптимизатора параметров ПИД-регулятора для управления нагревательным объектом* // Труды III Всероссийской научной конференции молодых ученых с международным участием «Теория и практика системного анализа», 21–24 мая 2014 г. –

- Т. 1. – Рыбинск: РГАТУ им. П.А. Соловьева, 2014. – С. 43–54.
3. ЕРЕМЕНКО Ю.И., ПОЛЕЩЕНКО Д.А., ГЛУЩЕНКО А.И. *Об особенностях практической реализации схемы ПИД-нейрорегулятора с самонастройкой для управления печами нагрева* // Приборы и системы. Управление. Контроль. Диагностика. – 2012. – №1. – С. 25–30.
  4. ЕРЕМЕНКО Ю.И., ПОЛЕЩЕНКО Д.А., ГЛУЩЕНКО А.И. *Об условиях применения ПИД-нейрорегулятора для управления объектами, описываемыми апериодическим звеном второго порядка с запаздыванием* // Приборы и системы. Управление. Контроль. Диагностика. – 2013. – №6. – С. 39–45.
  5. ЕРЕМЕНКО Ю.И., ПОЛЕЩЕНКО Д.А., ГЛУЩЕНКО А.И., ФОМИН А.В. *Об оценке применимости различных структур нейронной сети в реализации нейросетевого оптимизатора параметров ПИ-регулятора для управления тепловыми объектами* // Системы управления и информационные технологии. – 2014. – №3.2(57). – С. 236–241.
  6. ЗМЕУ К.В., МАРКОВ Н.А., ШИПИТЬКО И.А., НОТКИН В.С. *Безмодельное прогнозирующее инверсное нейрорегулирование с регенерируемым эталонным переходным процессом* // Информатика и системы управления. – 2009. – №3. – С. 109–117.
  7. КУДИНОВ Ю.И., КЕЛИНА А.Ю. *Упрощенный метод определения параметров нечетких ПИД регуляторов* // Мехатроника, автоматизация, управление. – 2013. – №1. – С. 12–22.
  8. ALEXANDROV A.G., PALENOV M.V *Self-tuning PID-I controller* // Proc. 18th IFAC World Congress, Milano, Italy, 2011. – P. 3635–3640.
  9. ANDERSON K.L., BLANKENSHIP G.I., LEBOW L.G. *A rule-based adaptive PID controller* // Proc. 27th IEEE Conf. Decision. Control, 1988. – P. 564–569.
  10. ASTROM K.J., HAGGLUND T., HANG C.C., HO W.K. *Automatic tuning and adaptation for PID controllers. A survey* // IFAC J. Control Eng. Practice. – 1993. – Vol. 1, №4. – P. 699–714.

11. CHEN J., HUANG T. *Applying neural networks to on-line updated PID controllers for nonlinear process control* // J. of Process Control. – 2004. – №14. – P. 211–230.
12. FANG M., ZHUO Y., LEE Z. *The application of the self-tuning neural network PID controller on the ship roll reduction in random waves* // Ocean Engineering. – 2010. – №37. – P. 529–538.
13. HORNIK K., STINCHCOMBE M., WHITE H. *Multilayer feed-forward networks are universal approximators* // Neural networks. – 1989. – №2(5). – P. 359–366.
14. NESLER C.G. *Experiences in applying adaptive control to thermal processes in buildings* // Proc. Amer. Control Conf., Boston, MA, 1985. – P. 1535–1540.
15. OMATU S., KHALID M., YUSOF R. *Neuro-Control and its Applications*. – London: Springer, 1995. – 255 p.
16. PFEIFFER B.-M. *Towards «plug and control»: self-tuning temperature controller for PLC* // Int. J. of Adaptive Control and Signal Processing. – 2000. – №14. – P. 519–532.
17. REYES J., ASTORGA C., ADAM M., GUERRERO G. *Bounded neuro-control position regulation for a geared DC motor* // Engineering Applications of Artificial Intelligence. – 2010. – №23. – P. 1398–1407.
18. TAN S.-H., HANG C.-C., CHAI J.-S. *Gain scheduling: from conventional to neuro-fuzzy* // Automatica. – 1997. – Vol. 33, №3. – P. 411–419.
19. ZHAO Z.Y., TOMIZUKA M., ISAKA S. *Fuzzy gain scheduling of PID controllers* // IEEE Transactions on systems, man, and cybernetics. – 1993. – Vol. 23, №5. – P. 1392–1398.
20. ZIEGLER J., NICHOLS N. *Optimum settings for automatic controllers* // Trans. ASME. – 1942. – №65. – P. 759–768.



## APPLYING NEURAL NETWORK-BASED TUNER TO OPTIMIZE PARAMETERS OF PI-CONTROLLER FOR HEATING FURNACE FUNCTIONING IN DIFFERENT MODES

**Yuri Eremenko**, Stary Oskol technological institute n.a. A.A. Ygarov (branch) NUST "MISIS", Stary Oskol, Doctor of Science, professor (erem49@mail.ru).

**Dmitry Poleshchenko**, Stary Oskol technological institute n.a. A.A. Ygarov (branch) NUST "MISIS", Stary Oskol, Cand.Sc, assistant professor (po-dima@yandex.ru).

**Anton Glushchenko**, Stary Oskol technological institute n.a. A.A. Ygarov (branch) NUST "MISIS", Stary Oskol, Cand.Sc (Stary Oskol, Makarenko microdistrict, 42, (4725)45-12-17, strondutt@mail.ru).

*Abstract: We propose a neural network-based tuner for online optimization of parameters of an automatic PI-controller for heating furnace control. The tuner consists of two neural networks responsible for adjusting coefficients  $K_P$  and  $K_I$  for furnace heating and cooling processes respectively. We develop a structure of a neural tuner and show by model experiments that such a tuner can be applied to control heating furnaces with the different value of the time constant. A muffle electric heating furnace functioning in different loading modes has been chosen as a plant. Having made our experiments, we conclude that such an optimizer helps to achieve about 23% decrease of time length and 19% decrease of energy consumption for each schedule in comparison with a conventional PI-controller.*

Keywords: neural network, adaptive control, PI-controller, neural tuner.

*Статья представлена к публикации  
членом редакционной коллегии Б.Т. Поляком*

*Поступила в редакцию 10.12.2014.  
Опубликована 31.07.2015.*