

УДК 681.515.8, 004.89

О МЕТОДЕ УПРАВЛЕНИЯ НАГРЕВАТЕЛЕМ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

© С. С. Абдуракипов, Е. Б. Бутаков

*Институт теплофизики им. С. С. Кутателадзе СО РАН,
630090, г. Новосибирск, просп. Академика Лаврентьева, 1
E-mail: s.s.abdurakipov@gmail.com*

Исследуется возможность применения нейронной сети долгой краткосрочной памяти (Long short-term memory, LSTM) для моделирования работы ПИД-регулятора. Реализован программный ПИД-регулятор для управления нагревателем с датчиком температуры на микроконтроллере Arduino, и разработана LSTM-модель, обученная на данных регулятора. Показано, что модель нейронной сети хорошо воспроизводит работу регулятора и может его полностью заменить при наличии многократно большего, но достаточного времени обработки данных. Продемонстрирована возможность использования данной модели в качестве детектора аномальной работы ПИД-регулятора.

Ключевые слова: системы управления, ПИД-регулятор, машинное обучение, рекуррентная нейронная сеть, детекция аномалий.

DOI: 10.15372/AUT20240213

EDN: KHVPAO

Введение. ПИД-регуляторы активно применяются в большинстве современных систем управления, например в системах управления летательными аппаратами, производственных процессах, таких как литьё металлов под давлением, химическое осаждение из паровой фазы, переработка электронных отходов, где требуется точный контроль температуры [1, 2]. ПИД-регуляторы реализуют механизм обратной связи контура управления (контроллера), работающего в пропорционально-интегрально-дифференциальном режиме. ПИД-регуляторы способны эффективно сглаживать любые колебания физических величин, возникающие в технической системе из-за различных внутренних и внешних факторов. Хотя математическое уравнение ПИД-регулятора относительно простое, применение показывает быстрые и стабильные результаты для большинства приложений [3, 4]. Разновидности и классификация ПИД-регуляторов подробно обсуждаются в работах [5, 6].

Общеизвестно, что ПИД-регулятор имеет три режима управления: пропорциональный, интегральный и дифференциальный. В пропорциональном режиме выходной сигнал прямо пропорционален величине ошибки. В интегральном — выходной сигнал зависит не только от величины ошибки, но и от её длительности во времени. В дифференциальном — режим управления по производной, в котором учитывается скорость изменения ошибки. Выходной сигнал представляет собой сумму всех трёх управляющих воздействий. Коэффициенты уравнения различаются для каждой конкретной задачи, поэтому выбор правильных параметров имеет решающее значение для стабильности управляемой системы [3]. Таким образом, поддерживается идеальный баланс, благодаря которому система может плавно достичь заданного значения из начального состояния без существенных колебаний. Необходимо настроить контроллер так, чтобы установка имела быстрый отклик, меньшее время нарастания и стабилизации, максимальную устойчивость и минимальное или полное отсутствие перерегулирования [4].

В научной литературе присутствует большое количество работ, посвящённых настройке, определению коэффициентов уравнения и оптимизации параметров

ПИД-регуляторов [7–11]. Существует множество подходов для поиска параметров ПИД-регуляторов [3, 4]. В литературе представлены различные методы получения оптимальных коэффициентов настройки регулятора, такие как метод Зиглера — Никольса [2, 3], метод оптимизации роя частиц [10], генетический алгоритм [11, 12] и др. За последнее десятилетие был совершён огромный скачок в области алгоритмов машинного обучения, и, естественно, он не обошёл стороной сферу промышленных систем управления [8], в том числе теплоэнергетику [13–15].

Развитие технологий привело к появлению современных систем управления, таких как MPC (model predictive control). Система MPC нацелена на вычисление управляющих воздействий путём решения задачи оптимизации в режиме реального времени. Однако нелинейность процессов, а также невыпуклость возникающей задачи оптимизации могут сделать онлайн-реализацию MPC-системы нетривиальной. Результаты [16] показывают, что имеются перспективы замены дорогостоящей в вычислительном отношении системы MPC гораздо более быстрой моделью машинного обучения. В связи с этим наиболее эффективными зарекомендовали себя подходы к моделированию контроллеров с использованием глубокого машинного обучения, такие как свёрточные и рекуррентные нейронные сети с механизмом внимания [17–20]. Наиболее современные варианты подобных моделей способны реализовывать адаптивную настройку параметров регулятора [21]. В данном случае будет рассмотрен подход к моделированию работы ПИД-регулятора на основе рекуррентной нейронной сети долгой краткосрочной памяти (Long short-term memory, LSTM).

Целью представленной работы являлись исследование применимости рекуррентных нейронных сетей для моделирования работы ПИД-регулятора на примере управления модельным нагревателем и разработка алгоритма для определения аномальных режимов работы ПИД-регулятора на основе обученной модели.

Описание инструментария.

ПИД-алгоритм управления нагревателем. Для экспериментов использовано простое устройство на микроконтроллере Arduino [22] с двумя нагревателями и двумя датчиками температуры. Подключение к компьютеру осуществлялось посредством USB-порта, нагреватели имели отдельный источник питания от сети. Регулирование мощности нагревателя и опрос датчиков температуры осуществлялись посредством реализованной программы на Python. В большей части экспериментов мы применяли только один нагреватель и один датчик температуры. На основе устройства был смоделирован процесс регулирования мощности нагревателя для соответствия заданному тестовому профилю температуры.

Выходной сигнал регулятора $U(t)$ определяется тремя слагаемыми: пропорциональным, интегральным и дифференциальным:

$$U(t) = u_{bias} + K_c e(t) + \frac{K_c}{\tau_I} \int e(t) dt + K_c \tau_D \frac{de(t)}{dt}. \quad (1)$$

На вход ПИД-регулятора подаются ошибка $e(t) = SP - PV$, которая соответствует разнице между заданным значением (SP) и измеряемой переменной процесса (PV), коэффициент усиления K_c , дополнительные параметры настройки для интегрального τ_I и дифференциального τ_D режимов управления, константа смещения u_{bias} .

Пропорциональное слагаемое регулирует выходной сигнал (мощность нагревателя) на основании величины ошибки $e(t)$ — отклонения температуры по датчику от установленного профиля. Чем больше величина отклонения (ошибки), тем выше уровень пропорционального сигнала. В то же время использование только пропорционального режима управления приводит к накоплению статической ошибки, которая появляется, когда выходной сигнал становится равным входному, при этом система ещё не успела достичь значения уставки. Чтобы уменьшить статическую ошибку, можно увеличить коэффициент

усиления K_c , однако, при большом значении данного коэффициента система, как правило, теряет устойчивость.

Для устранения статической ошибки вводится интегральное слагаемое. Оно пропорционально интегралу по времени от ошибки $e(t)$. Данная составляющая продолжает увеличивать выходной сигнал контроллера, чтобы температура на выходе нагревателя вернулась к заданному значению T_{SP} . Скорость интегрального действия имеет обратную зависимость от настройки контроллера τ_I .

Устройства, управляющие системой с помощью пропорционального и интегрального режимов, называются ПИ-регуляторами. Но зачастую ПИ-регуляторы реагируют на изменения достаточно медленно. В случае, когда необходимо более быстрое действие, дополнительно вводится дифференциальное слагаемое, пропорциональное скорости изменения ошибки. Вклад дифференциального управления также прямо пропорционален настройке регулятора τ_D . Вышеописанные режимы управления в сумме составляют основу работы ПИД-регулятора. Слагаемое u_{bias} — константа, которой обычно присваивается значение, соответствующее моменту первого переключения контроллера из ручного режима в автоматический. Это обеспечивает более плавный переход, если ошибка равна нулю при включении контроллера.

В представленной работе для моделирования ПИД-регулятора код реализован в программе Python с использованием класса `PidController`. В конструктор класса подаются следующие параметры: K_c , τ_I , τ_D , u_{bias} — пропорциональный, интегральный, дифференциальный коэффициенты и смещение; O_{min} , O_{max} — минимальное и максимальное значения выхода соответственно; `window` — ширина окна для усреднения ошибки (сглаживание при нежелательных осцилляциях на выходе ПИД-регулятора).

Программный модуль полностью реализует принцип работы ПИД-регулятора: считывание текущей температуры с сенсора и передача вычисленного значения мощности нагревателя обратно на устройство. Серия экспериментов проводилась на нескольких профилях температуры с несколькими подобранными параметрами регулятора методом Зиглера — Никольса [2, 3]. Далее в работе в качестве примера приводятся результаты для подобранных параметров регулятора: $u_{bias} = 0$, $K_c = 6$, $\tau_I = 75$, $\tau_D = 0$. Нижняя и верхняя границы уровня мощности нагревателя были выставлены 0 и 100. Результаты приводятся для указанных параметров регулятора в качестве наглядного примера для упрощения изложения.

В процессе экспериментов была реализована процедура хранения и предобработки данных в системе управления базами данных PostgreSQL. Для этого контроллер был запущен на 180 мин (одна серия измерений), чтобы сформировать достаточный массив данных для обучения модели. Длительность тестовой выборки составляла примерно 45 мин. Периодически изменялось заданное значение температуры, чтобы обеспечить хорошее сочетание установившегося и переходного режимов. Мы исходили из того, что 180 мин будет более чем достаточно для настройки нагревателя ПИД-регулятором так, чтобы температура соответствовала заданному температурному профилю T_{SP} . Вызывалась функция программного ПИД-регулятора на каждом временном шаге для расчёта выходной мощности нагревателя Q_{PID} , чтобы оставаться на требуемом уровне температуры. Обновление происходит раз в секунду. ПИД-регулятор не требует слишком больших вычислительных затрат, поэтому решение уравнения (1) можно легко получить за данное время.

На рис. 1 показаны результаты программного ПИД-контроллера в рамках одного цикла запуска. Как видно на рис. 1, регулятор успешно справляется со своей задачей поддержания температуры в соответствии с заданным профилем. Меняя параметры K_c , τ_I , τ_D , можно регулировать характеристики контроллера. Мы хотим, чтобы сплошная кривая с точкой, обозначающая датчик температуры, следовала за сплошной кривой, обозначающей

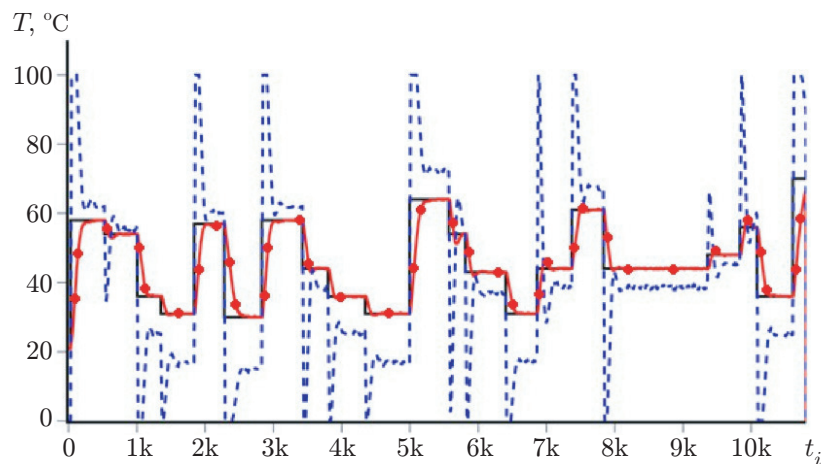


Рис. 1. Результаты программного ПИД-контроллера при следовании заданному профилю температуры. Сплошная кривая обозначает заданный профиль температуры T_{SP} , штриховая — контролируемую мощность нагревателя Q_{PID} , сплошная кривая с точкой — значения датчика температуры T_{PID}

заданное значение профиля температуры. Штриховой кривой показано, как необходимо отрегулировать мощность нагревателя, чтобы датчик соответствовал заданному значению.

Важно отметить, что в данной работе мы не фокусировались на тонкой настройке параметров ПИД-регулятора (ограничились базовой настройкой Зиглера — Никольса), а больше преследовали цель проверки гипотезы относительно применения методов машинного обучения для эффективной замены контроллера, в том числе для прогнозирования критических отклонений (аномалий).

Рекуррентная нейронная сеть. В представленной работе для моделирования ПИД-регулятора реализован класс LSTMController в программе Python. В качестве алгоритма машинного обучения была реализована рекуррентная нейронная сеть долгой краткосрочной памяти [23], которая хорошо себя показывает при работе с временными рядами. Отличие её от стандартной рекуррентной нейронной сети заключается в наличии ячеек памяти, которые помогают решить проблему исчезающего градиента [23]. Модель LSTM хорошо работает с проблемами описанного здесь типа, потому что контроллеру требуется помнить, что происходило ранее в системе управления (в интегральном режиме). В контексте имитации ПИД-регулятора в модель требуется ввести срез данных, таких как температура, уставка, ошибка и мощность нагревателя, и получить обратно то, каким должно быть следующее значение мощности нагревателя для достижения желаемого значения температуры. Это значение моделирует выходные данные, которые дал бы ПИД-регулятор с заданными параметрами. Важно отметить, что при изменении констант настройки регулятора используемая модель будет испытывать трудности, так как она обучалась на других данных.

Для обучения был использован набор данных из рис. 1: величина ошибки $T_{PID} - T_{SP}$, уставка и уровень мощности нагревателя Q_{PID} в качестве целевой переменной. Далее был сформирован набор данных, который представляет собой всевозможные наборы значений, в каждом из которых 15 последовательных значений (окно наблюдения) в качестве входа и следующее значение мощности в качестве выхода. Получившиеся датасеты были разделены на тренировочные (80 %) и валидационные (20 %). Тестовая выборка бралась из другого независимого эксперимента (запуска) с теми же настройками регулятора. Её размер совпадал с размером валидационной выборки.

Для реализации модели рекуррентной нейронной сети LSTM были использованы биб-

лиотеки Keras и TensorFlow. Ранее нами уже апробирован ряд алгоритмов машинного обучения (в том числе LSTM) [13–15]. Архитектура нейросети представляет собой два последовательных слоя LSTM из 100 вычислительных ячеек, слой Dropout (с вероятностью 5 %) после каждого рекуррентного слоя [23] и последний полносвязный (Dense) слой с 1 выходным нейроном для прогнозирования мощности нагрева. Для оптимизации весов нейросети применялся метод адаптивной инерции Adam [24]. В качестве функции потерь L использовалась среднеквадратичная ошибка (Mean squared error, MSE) [23]. На вход модели передаются последние 15 значений уставки и ошибки, на выходе получаем одно значение мощности нагревателя Q_{PID} . С помощью библиотеки Optuna подобраны следующие гиперпараметры: число слоёв, число ячеек в слое для наилучшей работы модели. Обучение проводилось за 300 эпох N_e с размером батча (пакета значений), равным 32. Также установлено условие ранней остановки (early stopping) [24]: обучение прерывается, если значение функции потерь не уменьшается в течение 25 эпох. Проверка на валидационной части выборки, не использованной для обучения, показала, что вычисления LSTM-модели хорошо приближают целевую функцию программного ПИД-регулятора (рис. 2).

Можно наблюдать резкое падение функции ошибки уже после нескольких эпох, а после 10 эпох происходит выход на плато. Функция ошибки на валидации также демонстрирует сходимость и находится чуть выше уровня кривой обучения. В свою очередь обучение останавливалось по условию ранней остановки при достижении примерно 100 эпох.

Результаты моделирования.

Сравнение нейронной сети и ПИД-регулятора. Для сравнения производительности программного ПИД-регулятора и нейросети проведена серия экспериментов с зафиксированным профилем температуры. Написана программа на Python для получения результатов работы нейросети в реальном времени по аналогии с принципом работы ПИД-регулятора, т. е. вычисление ошибки, передача данных в модель, получение на выходе значения мощности нагревателя и запись в результирующий массив. Здесь сразу заметен один недостаток использования LSTM-модели: необходимо иметь несколько предыдущих значений входных данных для корректной работы, так как она опирается на некоторый массив сохранённых данных. Поэтому на первых 15 шагах в модели задана комнатная температура (~ 20 °C). Результаты работы обеих программ на тестовом запуске можно увидеть на рис. 3. Здесь представлены 2 случая: управление нагревателем с помощью ПИД-регулятора и управление нейросетью.

Как можно заметить, результаты очень похожи, что говорит о том, что модель с хорошей точностью воспроизводит работу программного регулятора. Средняя абсолютная ошибка в прогнозировании мощности нагрева составляет порядка 1 %, а коэффициент детерминации — 99 %. Конечно, присутствуют небольшие отличия в сигналах, но они не существенны.

Таким образом, показано, что нейросеть может эффективно заменить ПИД-регулятор для контроля температуры в режиме реального времени путём управления мощностью нагревателя. Кроме того, работу модели можно корректировать различными способами, например, изменяя гиперпараметры и обучая модель на большем объёме данных. Стоит отметить, что полного совпадения результатов модели и ПИД-регулятора добиться сложно, однако, текущая точность уже является достаточной для корректной работы модели в реальных условиях.

Обнаружение аномалий. Как уже отмечалось, эффективность работы ПИД-контроллера зависит от набора параметров. В данной работе исследовался вопрос о влиянии резкого изменения параметров регулятора на качество прогнозирования моделью. Естественно, обученная на других значениях параметров LSTM-модель не сможет воспроизвести работу данного устройства с такой же низкой ошибкой. С одной стороны, это несомненно является недостатком, так как при смене параметров регулятора

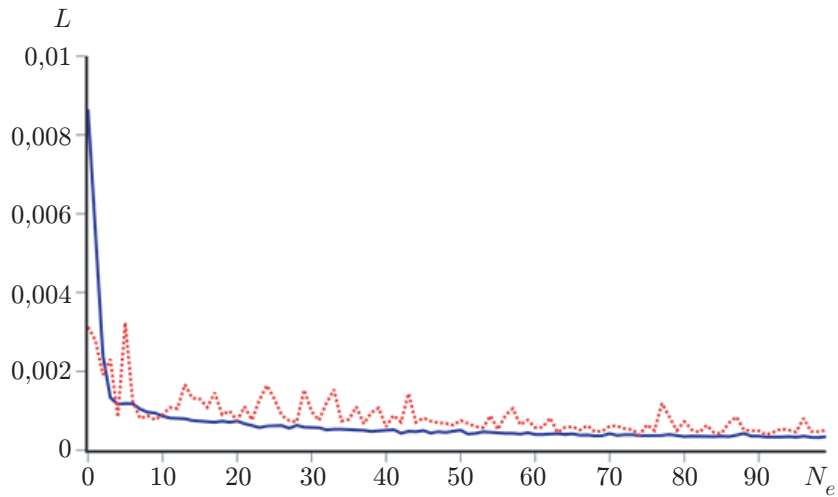


Рис. 2. Зависимость среднеквадратичной ошибки L от числа эпох N_e при обучении и валидации нейросети. Сплошная кривая — величина ошибки на обучающей выборке, пунктирная — величина ошибки на валидационной выборке

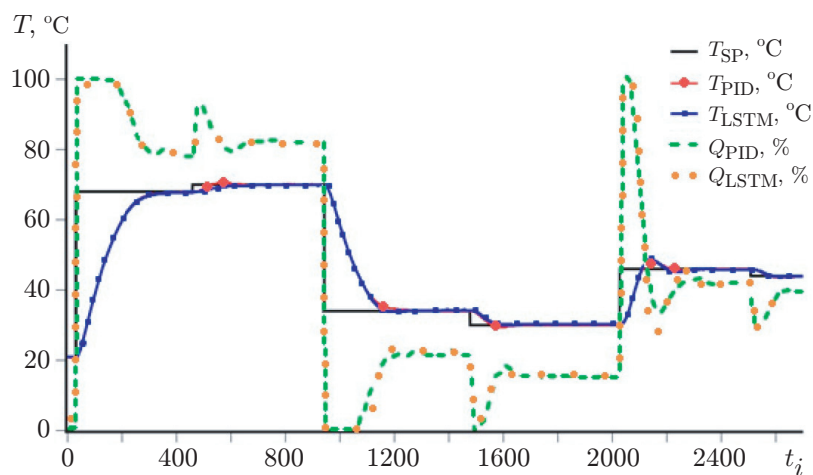


Рис. 3. Сравнение результатов программного ПИД-регулятора и рекуррентной нейросети при тестовом запуске. Сплошная кривая соответствует заданному профилю температуры T_{SP} ; штриховая кривая и точки — контролируемой мощности нагревателя, рассчитанной ПИД-регулятором Q_{PID} и нейросетью Q_{LSTM} соответственно; кривая с круглой точкой и кривая с квадратной точкой — показаниям датчика температуры в случае ПИД-управления T_{PID} и управления нейросетью T_{LSTM} соответственно

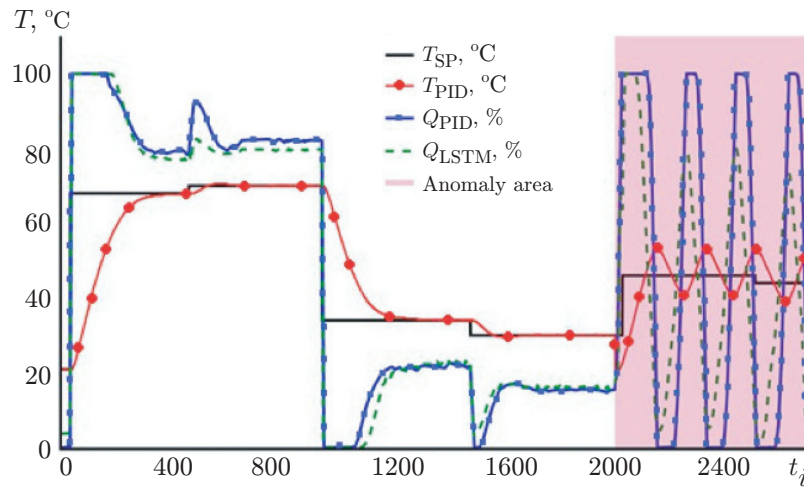


Рис. 4. Зависимости ПИД-регулятора и рекуррентной нейросети (LSTM) при мгновенном изменении параметров ПИД-регулятора (аномальный режим)

необходимо заново обучать нейросеть. С другой стороны, обычно параметры регулятора остаются фиксированными в процессе работы, а их изменение часто означает аномальную ситуацию, например поломку контроллера. Если при сравнении сигналов ПИД-регулятора и модели возникают сильные расхождения, то это означает неисправность контроллера. Таким образом, может быть разработан метод обнаружения аномалий в работе устройства.

Данный подход проверен на практике в рамках серии экспериментов. Был задан ступенчатый профиль температуры, и с некоторого момента времени (2000-й шаг) параметры ПИД-регулятора изменялись. Результаты одного из показательных запусков приведены на рис. 4. Сплошная кривая соответствует заданному профилю температуры T_{SP} ; кривая с квадратной точкой и штриховая кривая — контролируемой мощности нагревателя, рассчитанной ПИД-регулятором Q_{PID} и нейросетью Q_{LSTM} соответственно; кривая с круглой

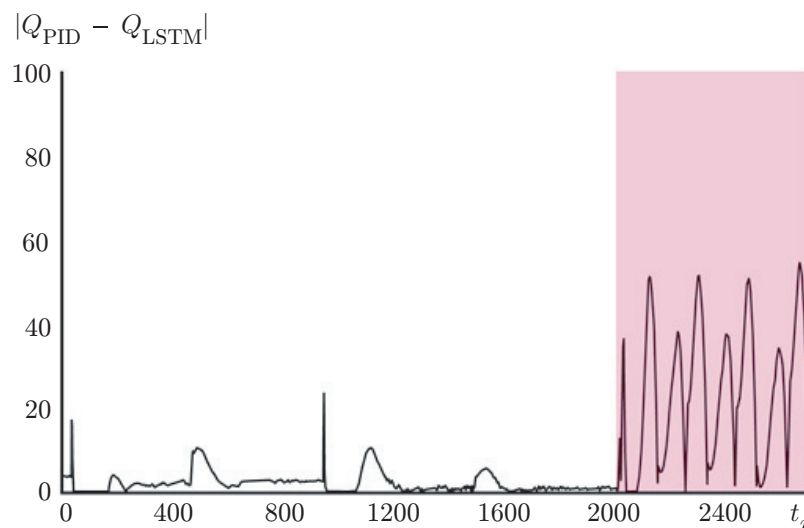


Рис. 5. График разности сигналов ПИД-регулятора Q_{PID} и нейросети Q_{LSTM} (контролируемая мощность нагревателя) при мгновенной смене параметров регулятора. Закрашенная область соответствует аномальным условиям (смена параметров ПИД-регулятора)

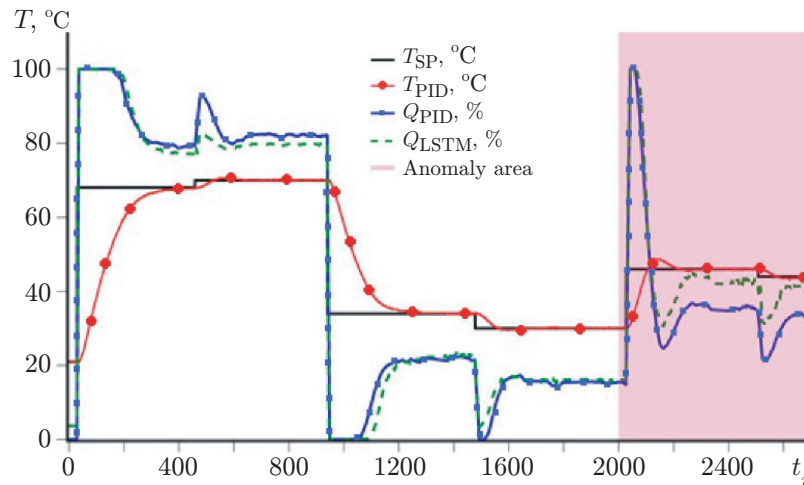


Рис. 6. Зависимости ПИД-регулятора и рекуррентной нейросети (LSTM) при резком изменении внешних условий

точкой — показаниям датчика температуры в случае ПИД-управления T_{PID} . Закрашенной областью отмечено изменение параметров контроллера.

Как и ожидалось, на начальном этапе сигналы очень похожи, но как только параметры регулятора меняются, наблюдаются сильные расхождения. Кроме того, поскольку именно ПИД-контроллер управляет нагревателем, то температура перестаёт соответствовать требуемому профилю. На рис. 5 приведён график разности сигналов регулятора и модели $|Q_{PID} - Q_{LSTM}|$ для обнаружения аномального поведения. Иметь подобную LSTM-модель в качестве резервного контроллера может быть полезно, так как при возникновении аномалий возможно переключение выхода с неисправного ПИД-регулятора на модель, имитирующую его поведение. В совокупности это может быть очень полезным для повышения стабильности работы системы.

В качестве одного из аргументов в пользу замены ПИД-регулятора на нейросеть в литературе часто упоминается меньшее время работы нейросети (инференса). В рамках работы была проверена данная гипотеза. При сравнении установлено, что время, затрачиваемое на одно вычисление модели, в 1000 раз больше, чем у регулятора, хотя и гораздо меньше 1 секунды. Это выглядит объяснимым, потому что одно из главных преимуществ ПИД-регулятора — это простота и скорость работы из-за решения несложного уравнения (1). Однако при применении более сложных контроллеров результат может быть другим.

В работе проверена гипотеза, может ли модель быть использована для обнаружения других типов аномалий, например изменения условий внешней среды или нарушения целостности конструкции устройства. С этой целью был добавлен второй нагреватель, который включался в определённый момент времени и работал на одной мощности. На первый взгляд, кажется очевидным, что модель не может качественно реагировать на изменение условий из-за того, что не обучалась на таких наблюдениях. С другой стороны, в реальных условиях столь сильное изменение внешних параметров означает аномальную ситуацию, приводящую к существенным отклонениям в выходном сигнале. На рис. 6 показаны результаты для случая внешнего нагревателя. Поведение кривых регулятора и модели в аномальной области выделено серым цветом.

Заключение. В данной статье исследована возможность моделирования работы ПИД-контроллера с помощью рекуррентной нейронной сети. Реализован функционал ПИД-контроллера на Python для управления Arduino устройством с двумя нагрева-

телями и двумя датчиками температуры воздуха. Рекуррентная нейронная сеть долгой краткосрочной памяти (LSTM) обучалась на данных, полученных с программного ПИД-контроллера, для моделирования его работы. Проведено сравнение выходных сигналов регулятора и модели на новых данных.

Показано, что модель достаточно точно воспроизводит поведение контроллера с небольшими отклонениями (в пределах 1 % выхода ПИД-регулятора и коэффициентом детерминации 99 %) и способна поддерживать заданный профиль температуры. Кроме того, исследованы возможности модели для обнаружения аномальных ситуаций, в том числе резкое изменение параметров ПИД-регулятора (его возможная поломка) и изменение внешних условий (сторонний источник нагрева). Показано, что модель может обнаруживать отклонения в настройках регулятора и служить двойником контроллера в случае выхода из строя последнего. При включении второго нагревателя существенных отклонений, как в случае изменения параметров регулятора, не зафиксировано.

Хотя модель может применяться в реальном времени, её недостатком является гораздо большее время работы (в 1000 раз) по сравнению с простыми вычислениями ПИД-регулятора. Однако разработанная модель может быть вполне актуальной при моделировании более сложных контроллеров.

СПИСОК ЛИТЕРАТУРЫ

1. **Visioli A.** Practical PID Control. London: Springer-Verlag London, 2006. 314 p.
2. **Johnson M. A., Moradi M. H.** PID Control. New Identification and Design Methods. London: Springer-Verlag London, 2005. 543 p.
3. **Tan K. K., Wang Q. G., Hang C. C., Hägglund T. J.** Advances in PID Control. London: Springer-Verlag London, 2012. 264 p.
4. **Востриков А. С., Французова Г. А.** Теория автоматического регулирования. Новосибирск: НГТУ, 2006. 368 с.
5. **Денисенко В. В.** ПИД-регуляторы: Принципы построения и модификации // Современные технологии автоматизации. 2006. № 4. С. 66–74.
6. **Александров А. Г., Паленов М. В.** Состояние и перспективы развития адаптивных ПИД-регуляторов // Автоматика и телемеханика. 2014. № 2. С. 16–30.
7. **Somefun O. A., Akingbade K., Dahunsi F.** The dilemma of PID tuning // Ann. Rev. in Control. 2021. **52**. P. 65–74.
8. **Borase R. P., Maghade D. K., Sondkar S. Y., Pawar S. N.** A review of PID control, tuning methods and applications // Int. Journ. Dynamics and Control. 2021. **9**. P. 818–827.
9. **Wang L.** PID Control System Design and Automatic Tuning Using MATLAB/Simulink. New Jersey (USA), Chichester (UK): John Wiley & Sons, 2020. 344 p.
10. **Sunghthong A., Assawinchaichote W.** Particle swam optimization based optimal PID parameters for air heater temperature control system // Proc. of the International Electrical Engineering Congress (iEECON 2016). Procedia Computer Science. Chiang Mai, Thailand, 2–4 March, 2016. Vol. 86. P. 108–111.
11. **Bagis A.** Determination of the PID controller parameters by modified genetic algorithm for improved performance // ISE. 2007. **23**, N 5. P. 1469–1480.
12. **Zhang J., Zhuang J., Du H., Wang S.** Self-organizing genetic algorithm-based tuning of PID controllers // Information Sciences. 2009. **179**, Iss. 7. P. 1007–1018.
13. **Abdurakipov S. S., Butakov E. B., Burdukov A. P. et al.** Using an artificial neural network to simulate the complete burnout of mechanoactivated coal // Combust. Explos. Shock Waves. 2019. **55**, N 6. P. 697–701.

14. **Абдуракипов С. С., Токарев М. П., Первунин К. С., Дулин В. М.** Моделирование характеристик тонального шума методами машинного обучения при обтекании гидропрофиля // Автометрия. 2019. **55**, № 2. С. 123–131. DOI: 10.15372/AUT20190213.
15. **Abdurakipov S., Butakov E.** Combustion anomalies detection for a thermal furnace based on Recurrent Neural Networks // Journ. Phys.: Conf. Ser. 2018. **1105**. 012043.
16. **Kumar S. S. P., Tulsyan A., Gopaluni B., Loewen P.** A deep learning architecture for predictive control // IFAC Papers Online. 2018. **51**, Iss. 18. P. 512–517.
17. **Zribi A., Chtourou M., Djemel M.** A new PID neural network controller design for nonlinear processes // Journ. Circuits, Systems and Computers. 2018. **27**, N 4. P. 1850065.
18. **Yoon J., Doh J.** Optimal PID control for hovering stabilization of quadcopter using long short-term memory // Adv. Eng. Inform. 2022. **53**. 101679. DOI: 10.1016/j.aei.2022.101679.
19. **Xie J., Yang R., Gooi H. B., Nguyen H. D.** PID-based CNN-LSTM for accuracy-boosted virtual sensor in battery thermal management system // Appl. Energy. 2023. **331**. 120424.
20. **Liang H., Sang Z.-K., Wu Y.-Z. et al.** High precision temperature control performance of a PID neural network-controlled heater under complex outdoor conditions // Appl. Therm. Eng. 2021. **195**. 117234. DOI: 10.1016/j.applthermaleng.2021.117234.
21. **Rodríguez-Abreo O., Rodríguez-Reséndiz J., Fuentes-Silva C. et al.** Self-tuning neural network PID with dynamic response control // IEEE Access. 2021. **9**. P. 65206–65215.
22. **Monk S.** Programming Arduino: Getting Started with Sketches. N.-Y.: McGraw-Hill Education, 2023. 162 p.
23. **Buduma Nit., Buduma Nik., Papa J.** Fundamentals of Deep Learning. 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2022. 390 p.
24. **Stevens E., Antiga L., Viehmann T.** Deep Learning with PyTorch. N.-Y.: Manning Publications, 2020. 520 p.

Поступила в редакцию 21.11.2023

После доработки 15.12.2023

Принята к публикации 26.12.2023
