

On a Method of Controlling Heater by Using Neural Networks

S. S. Abdurakipov^{1*} and E. B. Butakov¹

¹*Kutateladze Institute of Thermophysics, Siberian Branch, Russian Academy of Sciences,
Novosibirsk, 630090 Russia*

Received November 21, 2023; revised December 15, 2023; accepted December 26, 2023

Abstract—The possibility of using a long short-term memory (LSTM) neural network has been studied to simulate the operation of a PID controller. A programmable PID controller has been realized to control a heater with a temperature sensor on the basis of an Arduino microcontroller. A LSTM model trained on the controller data has been developed. It is shown that the neural network model accurately reproduces the operation of the controller and can completely replace it under the condition of a much greater but sufficient data processing time. The applicability of this model as a detector of abnormal operation of the PID controller is shown.

DOI: 10.3103/S8756699024700328

Keywords: *control systems, PID controller, machine learning, recurrent neural network, anomaly detection*

1. INTRODUCTION

PID controllers are actively used in the majority of modern control systems, for example, in the aircraft control systems, production process such as metal injection molding, chemical vapor deposition, electronic waste recycling, where an accurate temperature control is necessary [1, 2]. PID controllers realize the feedback mechanism of the control loop (controller) operating in the proportional-integral-differential (PID) mode. PID controllers can efficiently smooth any variations of physical quantities emerging in a technical system because of various internal and external factors. Even if the mathematical equation of a PID controller is relatively simple, its application demonstrates rapid and stable results for the majority of applications [3, 4]. The variety and classification of PID controllers are discussed in detail in [5, 6].

It is generally known that a PID controller has three control modes: proportional, integral, and differential. In the proportional mode, the output signal is directly proportional to the error value. In the integral mode, the output signal depends not only on the error value, but also on its duration in time. In the differential mode, the control is executed according to the derivative, where the rate of error change is considered. The output signal is a sum of all three control actions. The equation coefficients are different for each particular problem, therefore, the choice of correct parameters has a decisive importance for the stability of the controlled system [3]. So, an ideal balance is maintained, owing to which the system can smoothly reach the setpoint from its initial state with no significant oscillations. It is necessary to tune the controller so that the setup would have a quick response, smaller rise and stabilization times, maximum stability, and minimum retuning or its absence [4].

In the scientific literature, there are a lot of studies devoted to tuning, determining the coefficients of the equation and optimization of PID controller parameters [7–11]. There are numerous approaches to searching for parameters of PID controllers [3, 4]. In the literature, various methods for obtaining optimal coefficients of controller tuning are presented; for example, the Ziegler—Nichols method [2, 3], the particle swarm optimization method [10], genetic algorithm [11, 12], etc. Over the past decade, a huge jump was made in the domain of machine learning algorithms and, certainly, it did not bypass the sphere of technical control systems [8], including heat power engineering [13–15].

*E-mail: s.s.abdurakipov@gmail.com

Development of technologies has led to appearance of state-of-the-art control systems such as the model predictive control (MPC). The MPC system is aimed to calculate the control action by solving online optimization problems. However, the nonlinearity of the processes and nonconvexity of the emerging optimization problem can make the online realization of the MPC system nontrivial. The results obtained in [16] show that there are the prospects of replacement of the computationally costly MPC system with a faster machine learning model. In this connection, the most efficient approaches to modeling controllers with the use of deep machine learning are such as the convolutional and recurrent neural networks with an attention mechanism [17–20]. The most modern variants of such models are capable to realize the adaptive tuning of controller parameters [21]. In our case, we will consider the approach to modeling the PID controller operation based on the recurrent long short-term memory (LSTM) neural network.

The aim of this study is to investigate the applicability of recurrent neural networks for simulating the PID controller operation to determine anomalous modes of operation of a PID controller based on a trained model.

2. TOOLKIT DESCRIPTION

PID Algorithm of Heater Control. A simple device with an Arduino microcontroller [22] with two heaters and two temperature sensors was used for experiments. It was connected with a computer by means of an USB port; the heaters had a separate source of main power feed. Control of the heater power and reading the data from temperature sensors were performed by means of the software based on Python. In the majority of experiments, we used only one heater and one temperature sensor. Based on this device, the process of heat power control for its accordance with the test temperature profile was simulated.

The input signal of controller $U(t)$ is determined by three terms: proportional, integral, and differential, namely:

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

To the PID controller input, we put the error $e(t) = \text{SP} - \text{PV}$, which corresponds to the difference between the set point (SP) and the measured process variable (PV); the amplification coefficient K_c ; additional tuning parameters for the integral τ_I and differential τ_D control modes; and the bias constant u_{bias} .

The proportional term controls the input signal (the heater power) based on the value of the error $e(t)$ —the deviation of the temperature recorded by the sensor from the set profile. The larger the deviation (error), the higher the proportional signal level. At the same time, the use of the proportional mode alone leads to accumulation of the statistical error that appears when the output signal becomes equal to the input signal, at that, the system has not reached the set point value yet. To reduce the statistical error, it is possible to increase the amplification coefficient K_c ; however, at a larger value of this coefficient, the system usually loses its stability.

To eliminate the statistical error, the integral term is introduced. It is proportional to the time integral of the error $e(t)$. This component keeps increasing the output signal of controller to return the temperature at the heater output to the set point T_{SP} . The rate of the integral action has an inverse dependence on the controller tuning τ_I .

The devices that control a system by using the proportional and integral modes are called the PI controllers. However, it is often the case that PI controllers react quite slowly to changes. In the case when a more rapid action is required, one introduces additionally the differential term proportional to the rate of the error change. The differential control contribution is also directly proportional to the controller tuning τ_D . The above control modes in sum are the basis of the PID controller operation. The term u_{bias} is a constant to which one usually attributes the value corresponding to the time instant when the controller was switched from the manual to automatic mode for the first time. This provides a smoother transition, if the error is zero at the time instant when the controller is switched on.

In the presented study, the code for PID controller simulation is realized in Python with the use of the `PidController` class. To the class constructor input, the following parameters are put: K_c , τ_I , τ_D , u_{bias} are the proportional, integral, differential coefficients, and bias, respectively; O_{min} , O_{max} are the

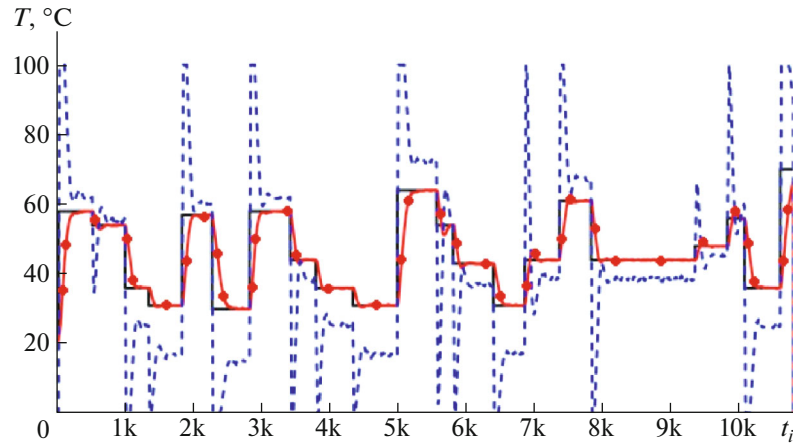


Fig. 1. The results of the programmable PID controller when following the set temperature profile. The solid curve refers to the set temperature profile T_{SP} ; the dotted line, to the controlled heater power Q_{PID} ; the solid curve with dots, to the values from the temperature sensor T_{PID} .

minimum and maximum output values, respectively; window parameter is the width of the window for error averaging (smoothing in the case of undesired oscillations at the PID controller output).

The software module completely realizes the principle of the PID controller operation: reading the current value of temperature from the sensor and transfer of the calculated value of the heater power back to the device input. A series of experiments was carried out at several temperature profiles and with several controller parameters fitted by the Ziegler–Nichols method [2, 3]. Below, as an example, we present the results for the fitted controller parameters: $u_{bias} = 0$, $K_c = 6$, $\tau_I = 75$, and $\tau_D = 0$. The lower and upper boundaries of the heater power level were set to be 0 and 100. The results are presented for the above parameters of controller as an illustrative example for the sake of simplicity.

In the course of experiments, the procedures of data storage and preprocessing were realized in the PostgreSQL database control system. For this purpose, the controller operated for 180 min (one series of measurements) to form a sufficient dataset for model training. The duration of the test sample was about 45 min. The set value of temperature was periodically changed to provide a good matching of the steady- and transient-state regimes. We proceeded from the idea that 180 min would be sufficient to tune the heater by using the PID controller so that the temperature would be in agreement with the set temperature profile T_{SP} . At each time step, the function of the programmable PID controller was called to calculate the heater output power Q_{PID} to keep the required temperature level. The update is performed once a second. The PID controller does not require too high computation costs, therefore, one can easily obtain the solution of Eq. (1) for this time interval.

Figure 1 shows the results of the programmable PID controller during one operation cycle. As is seen from Fig. 1, the controller succeeds in maintaining the temperature in accordance with the set profile. By varying parameters K_c , τ_I , and τ_D , it is possible to control the controller characteristics. Our aim is to make for the solid curve with dots to follow the solid curve that indicates the set value of the temperature profile. The dotted curve shows how it is necessary to control the heater power in order to provide that the temperature sensor data would be in agreement with the set point.

It is important to note that in this study, we were not focused on a fine tuning of the PID controller parameters (we did not go beyond the basic Ziegler–Nichols tuning), our main purpose was to check the hypothesis concerning application of the machine learning methods for effective replacement of the controller, including prediction of critical deviations (anomalies).

Recurrent Neural Network. In this study, to simulate the PID controller, the LSTMController class was realized in the Python code. As the machine learning algorithm, the long short-term memory recurrent neural network [23] was realized; it demonstrates its good performance when operating with time series. Its difference from the standard recurrent neural network is the presence of memory cells that help us solve the problem of vanishing gradients [23]. The LSTM model works well with the problems of described type, since it is necessary for the controller to remember what happened earlier in the control system (in the integral mode). In the context of PID controller imitation, it is necessary to introduce to

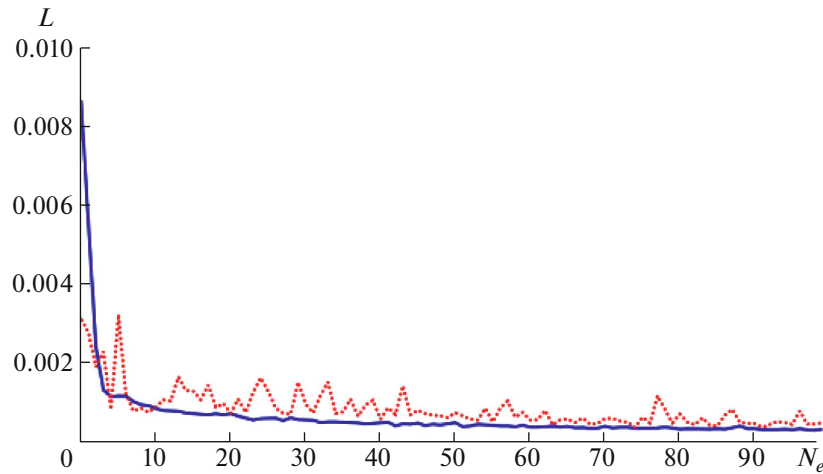


Fig. 2. Dependence of the mean squared error L on the number of epochs N_e during training and validation of the neural network. The blue curve is the magnitude of the error on the training set, the red curve is the magnitude of the error on the validation set.

the model the data such as the temperature, setpoint, error, and heater power and obtain the information concerning the next value of the heater power necessary to reach the desired value of temperature. This value models the output data, which would be yielded by the PID controller with the set parameters. It is important to note that if the constants of controller tuning change, the used model will have difficulties, since it was trained on other data.

For training, the data set from Fig. 1—the value of the error $T_{\text{PID}} - T_{\text{SP}}$, setpoint and heater power Q_{PID} —was used as the target variable. Then, the data set was formed, which consists of various sets of values, in each of them there are 15 consecutive values (the observation window) as the input data and the next value of the power as the output data. The data sets obtained were divided into the training (80%) and validation (20%) sets. The test sample was taken from another independent experiment (run) with the same tuned controller parameters. Its size coincided with that of the validation sample.

To realize the model of the LSTM recurrent neural network, the Keras and TensorFlow libraries were used. Earlier, we have already approved some machine learning algorithms (including LSTM) [13–15]. The neural network architecture includes two subsequent LSTM layers consisting of 100 computation cells, a Dropout layer (with a probability of 5%) after each recurrent layer [23], and the last fully connected (Dense) layer with 1 output neuron for prognosis of the heating power. To optimize the weights of the neural network, the adaptive inertia method Adam [24] was implemented. As loss function L , we used the mean squared error (MSE) [23]. The last 15 values of the setpoint and errors are put to the model input; at the output, we get one value of the heater power Q_{PID} . By using the Optuna library, the following hyperparameters were selected: the number of layers, the number of cells in a layer for the best operation of the model. Training was carried out for 300 epochs N_e with a batch size (the package of values) of 32. In addition, we established the early stopping condition [24]: the training process is interrupted if the value of loss function does not decrease during 25 epochs. The test on the validation part of the sample which was not used for training, has shown that the calculations of the LSTM model approximate well the target function of the programmable PID controller (Fig. 2).

One can see a sharp decay of the loss function already after several epochs; after 10 epochs, the loss function attains a plateau. The loss function on the validation sample exhibits its convergence and is slightly above the level of the training curve. In its turn, the training process was stopped according to the early stopping condition after approximately 100 epochs.

3. RESULTS OF SIMULATION

Comparison of the Neural Network and the PID Controller. To control the performance of the programmable PID controller with that of the neural network, a series of experiments with a fixed temperature profile was performed. The software program was written in Python to obtain online

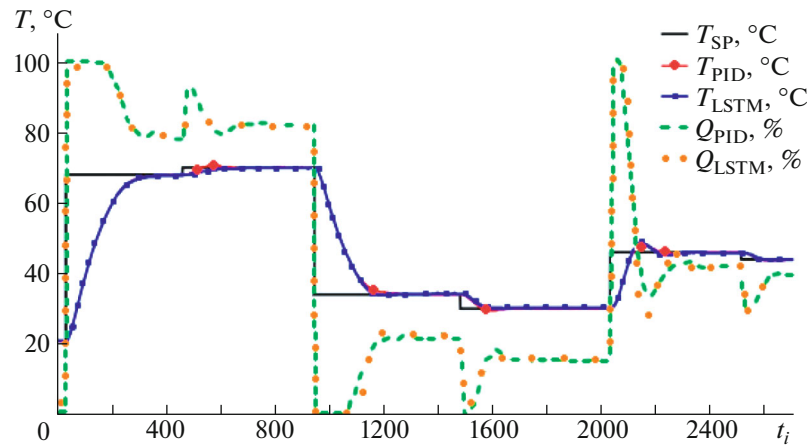


Fig. 3. Comparison of the results of the programmable PID controller and the recurrent neural network in the test run. The solid curve refers to the set temperature profile T_{SP} ; the dashed curve and dots, to the controlled heater power calculated by using the PID controller Q_{PID} and the neural network Q_{LSTM} , respectively; the curve with a round dot and the curve with a square dot refer to the data from the temperature sensor in the case of the PID control T_{PID} and the control using the neural network T_{LSTM} , respectively.

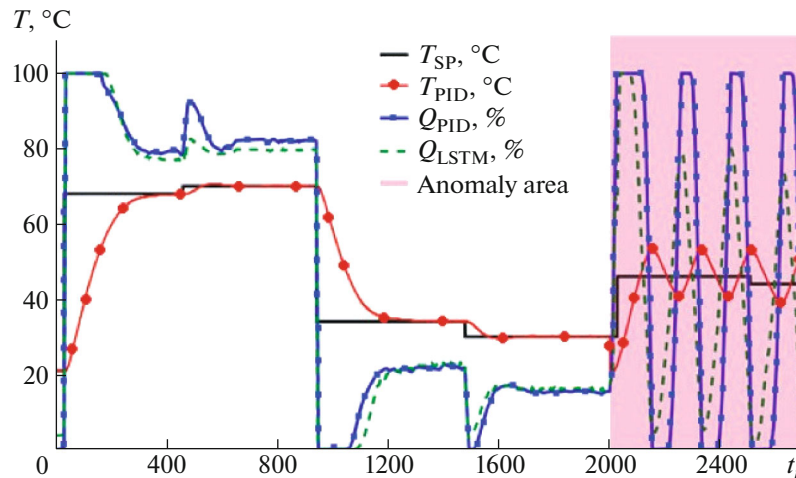


Fig. 4. Dependencies of the PID controller and recurrent neural network (LSTM) at an instantaneous change in the parameters of PID controller (anomalous mode).

the results of neural network operation by analogy with the principle of PID controller operation, i.e. calculation of the error, data transmission to the model, obtaining the heater power value at the model output and recording it to the resulting data array. Here, we immediately remark a disadvantage of the LSTM model: it is necessary to have several previous values of the input data for a correct operation, since the LSTM model is based on some array of stored data. Therefore, at the first 15 steps, the room temperature ($\sim 20^{\circ}\text{C}$) is set in the model. The results of operation of both programs in the test run are shown in Fig. 3. Here, two cases are presented: heater control by using the PID controller and the control by using the neural network.

As we can note, the results are similar, which is evidence that the model reproduces with a high accuracy the operation of the programmable controller. The mean absolute percentage error in prognosis of the heat power is about 1%; the determination coefficient is 99%. Of course, there are some slight differences in the signals, however, they are insignificant.

So, it is shown that the neural network can effectively replace the PID controller to control online temperature by means of controlling the heater power. In addition, the model operation can be corrected in different ways, for example, by changing the hyperparameters and training the model on a larger data

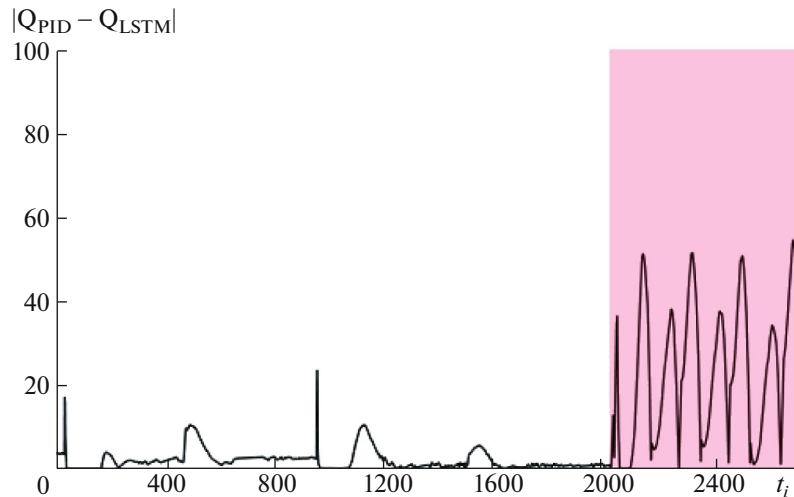


Fig. 5. Graph of the difference between the signals of the PID controller Q_{PID} and neural network Q_{LSTM} (controlled heater power) at an instantaneous change in the controller parameters. The fill area corresponds to anomalous conditions (change in the PID controller parameters).

set. It worth noting that it is difficult to achieve a complete coincidence of the results obtained by using the model and PID controller; however, the current accuracy is already sufficient for a correct model operation under real conditions.

Revealing Anomalies. As was already noted, the efficiency of PID controller operation depends on the set of parameters. In this study, the influence of abrupt changes in the controller parameters on the quality of model prognosis was investigated. Surely, the LSTM model trained on other values of parameters cannot reproduce the operation of this device with the same small error. On the one hand, this is its undeniable disadvantage, since in the case of changes in the controller parameters, it is necessary to retrain the neural network. On the other hand, usually, the controller parameters remain fixed in the process of its operation and their change often means that it is an anomalous situation, for instance, the controller failure. If there are significant discrepancies between the signals of the PID controller and the model, this means that the controller is faulty. Therefore, it is possible to develop a method for revealing anomalies in the device operation.

This approach was tested in practice in a series of experiments. A stepwise temperature profile was set; at some time instant (the 2000th step), the PID controller parameters were changed. The results of one of the demonstrative runs are presented in Fig. 4. The solid curve corresponds to the set temperature profile T_{SP} ; the curve with a square dot and the dashed curve, to the controlled heater power calculated by using the PID controller Q_{PID} and the neural network Q_{LSTM} , respectively; the curve with a round dot, to the data of the temperature sensor in the case of the PID control T_{PID} . The fill area indicates the change in the controller parameters.

As was expected, at the initial stage, these signals are similar; however, as soon as the controller parameters change, significant discrepancies are observed. In addition, since it is the PID controller that controls the heater, then the temperature already does not satisfy the required profile. Figure 5 shows the graph of the difference between the controller and model signals $|Q_{PID} - Q_{LSTM}|$ to reveal the anomalous behavior. It can be useful to have such a LSTM model as a standby controller, since if anomalies occur, it is possible to switch the output from the faulty PID controller to the model simulating its behavior. This combination can be very useful to enhance the stability of the system operation.

One of the arguments in favor of replacement of the PID controller with the neural network, which is often mentioned in scientific literature, is a smaller operation time of the neural network (inference). This hypothesis was verified in our work. In the process of comparison, it was established that the time necessary for one calculation of the model is by 1000 times larger than that of the controller, even if it is much smaller than 1 second. This fact seems to be explicable, since one of the main advantages of PID controller is its simplicity and operation rate because of solving a simple equation (1). However, when using more complicated controllers, the result may be other.

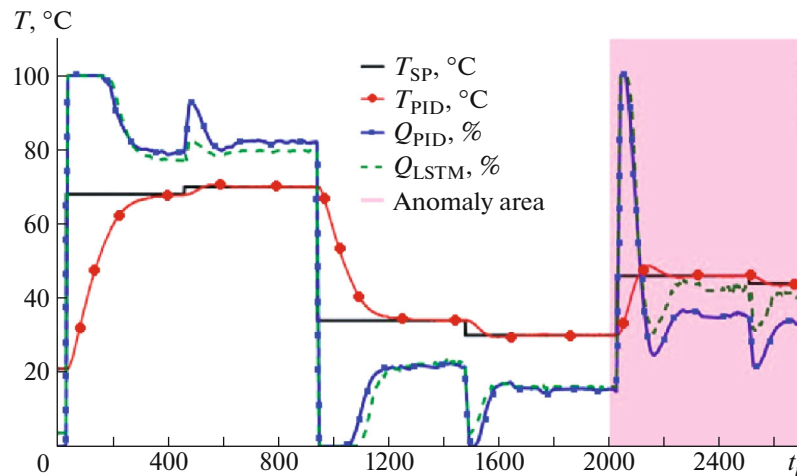


Fig. 6. Dependencies of the PID controller and recurrent neural network (LSTM) at abrupt changes in the external conditions.

In this study, we have verified the hypothesis if the model can be used to reveal other types of anomalies, for example, changes in the environment conditions or violation of the integrity of the device structure. With this purpose, we added the second heater, which was switched on at some time instant and operated at the same power. At first glance, it seems to be obvious that the model cannot exhibit a high performance concerning its reaction to changes in the conditions, since it was not trained on such observations. On the other hand, in the reality, such a significant change in the external parameters means an anomalous situation leading to significant deviations in the output signal. Figure 6 shows the results for the case of the external heater. The behavior of the curves that refer to the controller and model in the anomalous area is indicated with a gray color.

4. CONCLUSIONS

In this study, the possibility of simulation of the PID controller operation with the use of the recurrent neural network is investigated. The functional of the PID controller is realized in Python to control the Arduino device with two heaters and two sensors of air temperature. The recurrent long short-term memory (LSTM) neural network was trained on the data obtained from the programmable PID controller, to simulate its operation. The output signals of the controller and model were compared on the set of new data.

It was shown that the model reproduces the controller behavior with a sufficient accuracy with some small deviations (within 1% of the PID controller output and a determination coefficient of 99%) and is capable of maintaining a set temperature profile. In addition, the possibilities of the model to reveal anomalous situations, including an abrupt change in the PID controller parameters (its possible failure) and a change in the external conditions (an external source of heating) were investigated. It was shown that the model can reveal deviations in the tuned controller parameters and serve as a duplicate of the controller in the case of failure of the latter. When switching on the second heater or changing the controller parameters, no significant deviations were observed.

Despite the fact that the model can be used online, its disadvantage is a significantly larger operation time (by 1000 times) in comparison with simple calculations of the PID controller. However, the developed model can be quite relevant when simulating more complicated controllers.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they do not have a conflicts of interest.

REFERENCES

1. A. Visioli, *Practical PID Control*, Advances in Industrial Control (Springer, London, 2006). <https://doi.org/10.1007/1-84628-586-0>
2. J. Crowe, G. R. Chen, R. Ferdous, D. R. Greenwood, M. J. Grimble, H. P. Huang, J. C. Jeng, M. A. Johnson, M. R. Katebi, S. Kwong, et al., *PID Control: New Identification and Design Methods*, Ed. by M. A. Johnson and M. H. Moradi (Springer, London, 2005). <https://doi.org/10.1007/1-84628-148-2>
3. K. K. Tan, Q. G. Wang, C. C. Hang, and T. J. Haggglund, *Advances in PID Control*, Advances in Industrial Control (Springer, London, 2012). <https://doi.org/10.1007/978-1-4471-0861-0>
4. A. S. Vostrikov and G. A. Frantsuzova, *Theory of Automatic Control* (Novosibirsk. Gos. Tekh. Univ., Novosibirsk, 2006).
5. V. V. Denisenko, "PID controllers: Principles of construction and modifications," *Sovrem. Tekhnol. Avtom.*, No. 4, 66–74 (2006).
6. A. G. Alexandrov and M. V. Palenov, "Adaptive PID controllers: State of the art and development prospects," *Autom. Remote Control* **75**, 188–199 (2014). <https://doi.org/10.1134/S0005117914020027>
7. O. A. Somefun, K. Akingbade, and F. Dahunsi, "The dilemma of PID tuning," *Annu. Rev. Control* **52**, 65–74 (2021). <https://doi.org/10.1016/j.arcontrol.2021.05.002>
8. R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control* **9**, 818–827 (2021). <https://doi.org/10.1007/s40435-020-00665-4>
9. L. Wang, *PID Control System Design and Automatic Tuning Using MATLAB/Simulink* (Wiley, Hoboken, N.J., 2020). <https://doi.org/10.1002/9781119469414>
10. A. Sunghong and W. Assawinchaichote, "Particle swarm optimization based optimal PID parameters for air heater temperature control system," *Procedia Comput. Sci.* **86**, 108–111 (2016). <https://doi.org/10.1016/j.procs.2016.05.027>
11. A. Bagis, "Determination of the PID controller parameters by modified genetic algorithm for improved performance," *J. Inf. Sci. Eng.* **23**, 1469–1480 (2007). <https://doi.org/10.6688/JISE.2007.23.5.11>
12. J. Zhang, J. Zhuang, H. Du, and S. Wang, "Self-organizing genetic algorithm based tuning of PID controllers," *Inf. Sci.* **179**, 1007–1018 (2009). <https://doi.org/10.1016/j.ins.2008.11.038>
13. S. S. Abdurakipov, E. B. Butakov, A. P. Burdukov, A. V. Kuznetsov, and G. V. Chernova, "Using an artificial neural network to simulate the complete burnout of mechanoactivated coal," *Combust., Explos., Shock Waves* **55**, 697–701 (2019). <https://doi.org/10.1134/s0010508219060108>
14. S. S. Abdurakipov, M. P. Tokarev, K. S. Pervunin, and V. M. Dulin, "Modeling of the tonal noise characteristics in a foil flow by using machine learning," *Optoelectron., Instrum. Data Process.* **55**, 205–211 (2019). <https://doi.org/10.3103/S8756699019020134>
15. S. Abdurakipov and E. Butakov, "Combustion anomalies detection for a thermal furnace based on Recurrent Neural Networks," *J. Phys.: Conf. Ser.* **1105**, 012043 (2018). <https://doi.org/10.1088/1742-6596/1105/1/012043>
16. S. S. P. Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen, "A deep learning architecture for predictive control," *IFAC-PapersOnLine* **51**, 512–517 (2018). <https://doi.org/10.1016/j.ifacol.2018.09.373>
17. A. Zribi, M. Chtourou, and M. Djemel, "A new PID neural network controller design for nonlinear processes," *J. Circuits, Syst. Comput.* **27**, 1850065 (2018). <https://doi.org/10.1142/s0218126618500652>
18. J. Yoon and J. Doh, "Optimal PID control for hovering stabilization of quadcopter using long short term memory," *Adv. Eng. Inf.* **53**, 101679 (2022). <https://doi.org/10.1016/j.aei.2022.101679>
19. J. Xie, R. Yang, H. B. Gooi, and H. D. Nguyen, "PID-based CNN-LSTM for accuracy-boosted virtual sensor in battery thermal management system," *Appl. Energy* **331**, 120424 (2023). <https://doi.org/10.1016/j.apenergy.2022.120424>
20. H. Liang, Z. Sang, Yu. Wu, Yo. Zhang, and R. Zhao, "High precision temperature control performance of a PID neural network-controlled heater under complex outdoor conditions," *Appl. Therm. Eng.* **195**, 117234 (2021). <https://doi.org/10.1016/j.applthermaleng.2021.117234>
21. O. Rodríguez-Abreo, J. Rodríguez-Reséndiz, C. Fuentes-Silva, R. Hernández-Alvarado, and M. D. C. P. T. Falcón, "Self-tuning neural network PID with dynamic response control," *IEEE Access* **9**, 65206–65215 (2021). <https://doi.org/10.1109/ACCESS.2021.3075452>
22. S. Monk, *Programming Arduino: Getting Started with Sketches* (McGraw-Hill, New York, 2023).
23. N. Buduma, N. Buduma, and J. Papa, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms* (O'Reilly Media, Sebastopol, Calif., 2022).
24. E. Stevens, L. Antiga, and T. Viehmann, *Deep Learning with PyTorch* (Manning Publications, Shelter Island, N.Y., 2020).

Translated by E. Smirnova

Publisher's Note. Allerton Press, Inc. remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.