Time-Delay Temperature Control System Design based on Recurrent Neural Network

Yuan Liu
Division of Electronics and Informatics
Gunma University
Kiryu, Japan
t192d005@gunma-u.ac.jp

Song Xu
Department of Electronics and Informatics
Jiangsu University of Science and Technology
Zhenjiang, China
xusong0922@outlook.com

Shinya Kobori
Division of Electronics and Informatics
Gunma University
Kiryu, Japan
t181d026@gunma-u.ac.jp

Seiji Hashimoto
Division of Electronics and Informatics
Gunma University
Kiryu, Japan
hashimotos@gunma-u.ac.jp

Takahiro Kawaguchi
Division of Electronics and Informatics
Gunma University
Kiryu, Japan
kawaguchi@gunma-u.ac.jp

2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) | 978-1-7281-6207-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPS49255.2021.9468261

***Abstract*—A Recurrent Neural Network (RNN) is a special neural network sequence model that is very suitable for dealing with time series tasks. In various industrial processing systems, it has achieved good performances. In this paper, a RNN model which is driven by an ideal reference model is proposed for the single-input single-output(SISO) temperature control system with time-delay. An ideal reference model is introduced to provides a more valuable teaching signal for helping RNN controller to obtain higher learning efficiency and providing suitable control input to the temperature control system. Meanwhile, Adam optimization algorithm which can get adaptive learning rates is used to update parameters and improve the control performance of the RNN. Further, a classical integral proportional derivative (I-PD) controller is designed to reduce the effects caused by the temperature setting value kick during the RNN learning period. Simulations were developed under the MATLAB environment to evaluate the proposed control system performance. In order to demonstrate the efficiency and application of the proposed RNN control method, the simulation results based on the actual temperature model are compared quantitatively.**

***Keywords*—Recurrent Neural Network, Adam optimization algorithm, I-PD controller, temperature control**

I. Introduction

Effective and precise temperature control is vital in industrial processes. It seriously affects the product quality, production costs and energy consumption. Temperature controllers are widely used in thermal processes, which can effectively adjust process temperatures of the manufacturing process. For getting higher control precision and ensure quick and stable transient response of the temperature system, various control strategies have been compared and applied [1], [2]. In the single-input single-output temperature control systems, the classical proportional-integral-derivative (PID) control is universally accepted in industrial production process for its simple implementation and reliability [3].

However, most industrial processes are nonlinear and have large time delays. A conventional PID control always fails to perform well and is more and more difficult to meet the needs in practical applications. Thus, a fuzzy-proportional integral control method targeting the limitations of conventional PID control has been proposed and solved, achieving much better control performances during the overall system response process [4]. Meanwhile, considering the control system modeling error, a precise mathematical model of the controlled plant is always necessary for better performances. A variety of identification methods have been developed to structure models as accurate as possible based on the step response data [5]. To ensure the stability and robustness of temperature control systems, advanced compensation methods have been applied, such as Smith estimator predictive control [6] and model predictive control [7].

Considering the problem that parameters are always fixed in typical heating system control processing, it's difficult to adjust parameters in operation accurately in accordance to the actual situation. The disturbance in operation may greatly deteriorate the performance of systems and even make the system unstable. Neural networks have a wide range of applications in solving different kinds of complex problems, especially in multivariable and nonlinear systems. They can self-adjust their parameters timely to solve problems caused by changes in the controlled system. Over the past few decades, Neural networks have successfully contributed to the single-input single-output temperature control system [8]–[10]. For a nonlinear thermal system with large delay time, an adaptive control system can effectively adapt itself to various changing conditions, reducing the undesired overshoot and improving the system response performance [11], [12]. Furthermore, to improve the hyper-parameter training efficiency of neural networks, various optimization methods have been proposed such as Stochastic Gradient Descent (SGD) and Adam optimization algorithm [13]. In addition, the activation functions such as *Tanh*, *Sigmoid* and *ReLu* have been greatly developed [14].

In our previous work, we proposed a reference-model-based neural network control strategy [15]. Based on the research, a recurrent neural network model combined with an ideal reference model is proposed to improve the transient response and steady state response of the large time-delay temperature system. In this method, an ideal reference model based on the real temperature plant is introduced, the squared errors between the ideal output and the actual output are calculated as the teaching signals of RNN controller. It online adjusts parameters and obtain optimal control outputs, the Adam optimization algorithm is used to improve the performance of the RNN. In addition, a feedforward controller is introduced to provide an extra input for the RNN to learn and adjust its hyper-parameters more effectively. The RNN model is driven by the reference model provided teaching signal and its control output is together with the I-PD controller output make accurate adjustments of the system input signals. The rest of this paper is as follows: Section II presents the overall framework of the proposed RNN control method. Simulation experiments and analyses of the proposed method are presented in Sections III. And also, the I-PD control method is used for comparison and verifying the proposed RNN control performance. A brief conclusion is given in Section IV.

II. Configuration of RNN Controller with a Reference Model

This section introduces the framework of the proposed SISO RNN temperature control system which is driven by an ideal reference model.

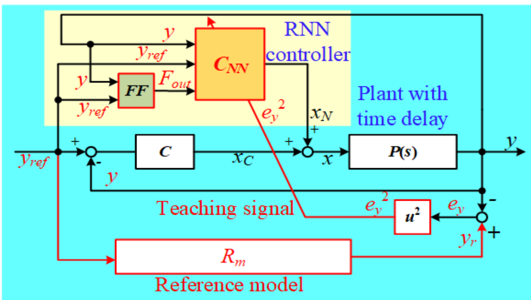


Fig. 1. The architecture of the proposed RNN control system

In Fig. 1, y_{ref} is the target value (reference model output) for the system, e_y indicates the error calculated by the ideal reference model R_m output and the real temperature output y. Here, the squared error signal ${e_y}^2$ is the self-learning signal of the proposed RNN controller C_{NN}. It can speed up the RNN learning progress during the temperature rising period, and also make the actual output track the target model output faster and smoother near the target value. *FF* is the feedforward compensator that is a proportional-derivative (PD) controller. In this proposed method, the output of the feedforward compensator is added to the control input signal of the RNN control for improving the learning efficiency of the RNN controller. Here, C represents the I-PD controller and the control input of the control system x consists of x_N (RNN control output) and x_C (I-PD control output). R_m has the same delay time of controlled object *P(s)* and provide the ideal temperature output. It is a predesigned model according to the real plant. The detailed description is mainly separated into four main parts.

A. Time-delay temperature plant and I-PD control compensation

The controlled temperature plant can be represented in a first-order plus time-delay model for its characteristics. The transfer function of the controlled plant is as (1), where τ is time delay, T and K are time constant and steady state gain.

$$P(s) = \frac{K}{Ts+1}e^{-\tau s} \tag{1}$$

In order to avoid the influences of temperature setting value kick before the RNN controller completes learning and mainly acts, the feedback controller C in Fig. 1 that is a I-PD controller is applied to help control during RNN learning initial state. The structure of the I-PD controller C is designed as Fig. 2, T_i, T_d and K_p are integral constant, differential constant and the proportional gain. The low-pass filter gain η is for reducing the high-frequency gain and noise.

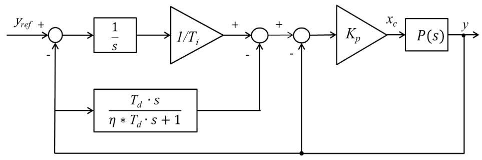


Fig. 2. Structure of conventional I-PD controller

In actual application, Ziegler-Nichols tuning rules are the most widely applied method of tuning the PID controller parameters for its stability and applicability have been proven [16]. Hence, the I-PD parameters K_p, T_i and T_d determined by the controlled plant and can be deduced from (2), respectively.

$$K_p = 1.2\frac{T}{\tau}; T_i = 0.5T; T_d = 2\tau \tag{2}$$

B. RNN controller with Adam optimization algorithm

In our proposed RNN control system, the number of neurons in the input layer is three, in the hidden layer is ten and in the output layer is one.

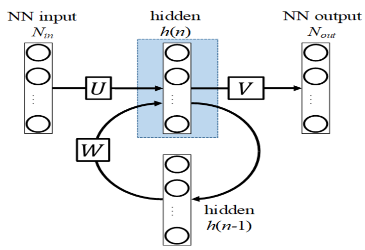


Fig. 3. Structure of the proposed recurrent neural network controller

In our proposed RNN method, the ideal model output y_{ref}, the actual system output y and the feedforward compensator output F_{out} are the three inputs of the RNN controller. The internal calculation direction of RNN control is from the input neurons N_{in} to output neurons N_{out} is as Fig. 4. We can get the output of the RNN controller x_N. Here, the *U* is the weight of input neurons, *V* is the weight of output neurons and *W* is the weight of the state memory neurons of RNN controller. And the network activation function is *f(·)*. The output of the RNN controller is given in (3), where N_{in} and N_{out} represent the input vector and output vector, respectively. Here, *b* and *c* represent the hidden layer bias and the output layer bias, respectively.

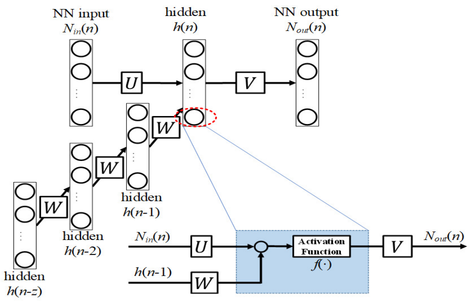


Fig. 4. Calculation process for RNN controller

$$N_{out} = V * f(f(U * N_{in}) + W * h(t-1) + b) + c \tag{3}$$

The Back Propagation Through Time (BPTT) algorithm is applied to update the parameters of RNN controller as shown in Fig. 5. At n^{th} iteration of RNN, the state memory layer can store z steps of the previous calculated data.

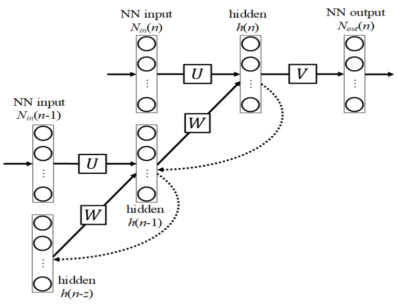


Fig. 5. BPTT: Back Propagation Through Time

Then the output layer of the RNN controller start back propagation to update parameters. The error signal is used for loss function calculation and at n^{th} iteration the gradient error signal of neural network is derived from (4), where *m* means the number of the output neurons.

$$E = \frac{1}{2}\sum_{i=1}^{m}(y_r(n) - y(n))^2 \tag{4}$$

The local gradient for the output layer and hidden layer can be calculated as (5) and (6) based on the gradient error, respectively. In (7) and (8), *u(n)* and *o(n)* are the induced local domain of the hidden and output layers, respectively.

$$\delta_h(n) = \frac{\partial E}{\partial u(n)} = f^{'}(u(n)) * V^T \delta_o(n) \tag{5}$$

$$\delta_o(n) = \frac{\partial E}{\partial o(n)} = f^{'}(o(n)) * (y(n) - t(n)) \tag{6}$$

$$u(n) = U * N_{in}(n) + W * h(n-1) + b(n) \tag{7}$$

$$o(n) = V * h(n) + c(n) \tag{8}$$

For the z steps stored calculation neurons, the local gradient from n^{th} iteration to *(n-z)*th iteration can be calculated by (9).

$$\delta_h(n-z-1) = \delta_h(n-z) * (W * f^{'}(u(n-z-1))) \tag{9}$$

According to the BPTT algorithm, the update of the weight *U* of the input layer can be calculated as (10), and the update of the weight *V* of the output layer can be calculated as (11), the weight corrections of the hidden layer and the memory neurons *W* is expressed as (12), where α is the weight training gain, and can be optimized by *Adam* algorithm, which is well-suited for handling problems with large amounts of data for its higher computational efficiency [17].

$$U(n+1) = U(n) - \alpha \sum_{i=0}^{z} \delta_h(n-i) * N_{in}(n-i) \tag{10}$$

$$V(n+1) = V(n) - \alpha \sum_{i=0}^{z} \delta_o(n) * h(n) \tag{11}$$

$$W(n+1) = W(n) - \alpha \sum_{i=0}^{z} \delta_h(n-z) * h(n-z-1) \tag{12}$$

The hidden layer neuron bias *b* and output layer neuron bias *c* are adjusted by local gradient of hidden layer δ_h*(n)* and output layer δ_o*(n)*, respectively, as given in (13) and (14), where β is the bias.

$$b(n+1) = b(n) - \beta\Delta b = b(n) - \beta \sum_{i=0}^{z} \delta_h(n-z) \tag{13}$$

$$c(n+1) = c(n) - \beta\Delta c = c(n) - \beta \sum_{i=0}^{z} \delta_o(n) \tag{14}$$

Based on the updated neuron weights and bias obtained from the above calculation, to improve the learning efficiency of the RNN, the *Adam* optimization is used to hold the term which is the exponentially damped average of the past slop square *v(n)* and *m(n)* at n^{th} iteration. Assuming that *g(n)* is the gradients of the neuron at n^{th} iteration (in this situation, it's the teaching signal ${e_y}^2$), the calculation of the updating biased first moment and second raw moment estimation are as (15) and (16), respectively. Here, β_1 and β_2 are the hyper-parameters of *Adam*.

$$m(n) = \beta_1 * m(n-1) + (1-\beta_1) * g(n) \tag{15}$$

$$v(n) = \beta_2 * v(n-1) + (1-\beta_2) * g^2(n) \tag{16}$$

Then, the calculated bias-corrected first moment and second raw moment estimation can be expressed as (17) and (18), respectively. The final correction of the Δ *w(n)* is given in (19), where *w(n)* represents the weight matrix of one layer, α_{ad} is the step size of *Adam* calculation, ε is the hyper-parameter of

Adam. In this way, the weights U, V and W are updated. The *ReLu* neuron activation function is applied to improve learning efficiency in the proposed RNN controller.

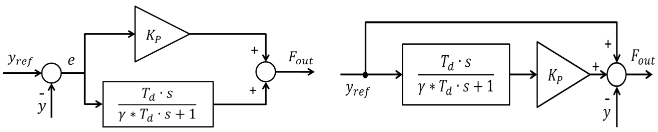
$$\hat{m}(n) = \frac{m(n)}{1-\beta_1^n} \tag{17}$$

$$\hat{v}(n) = \frac{v(n)}{1-\beta_2^n} \tag{18}$$

$$\Delta w(n) = -\frac{\alpha_{ad}}{sqrt\hat{v}(n)+\varepsilon} * \hat{m}(n) \tag{19}$$

C. Feedforward compensator

In order to ensure efficient self-study of the RNN controller, the feedforward compensator *FF* is added to provide an extra input of the RNN controller. In this paper, two types compensators: PD(proportional-derivative) and 2DOF (two degree of freedom) are considered in Fig. 6(a) and (b), respectively, the parameters K_p, T_d are obtained as same as that of I-PD controller as introduced above, γ is the low-pass filter (LPF) gain of the differential term, in this case γ = 0.5.



(a) Proportional-derivative structure (b) Two degree of freedom structure

Fig. 6. Structures of Feedforward compensator

D. Ideal reference model design

In this paper, a reference model in the form of the transfer function of the control plant is designed to provide the desired control output and learning signal to control system. The ideal model is designed based on the introduced controlled object model. The delay time τ is approximated by a second-order rational function as (20). Here, a gain R is designed to obtain faster transient response of the controlled object.

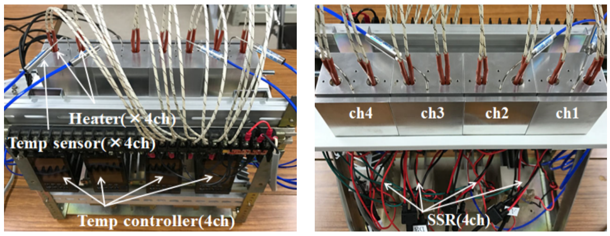
$$R(s) \approx \frac{K}{T*R*s+1} \times \frac{1}{(\frac{\tau s}{2}+1)^2} \tag{20}$$

III. Simulation Results

A. Experimental setup and system identification

The experimental setup is presented in Fig. 7. The experiment control system consists of 4 coupled blocks and each block has two heaters and one sensor for detecting temperature. The temperature of each block (0-400 [°C]) can be transformed into the output voltage (0-10 [V]). The two heaters of the block are driven by pulse width modulation (PWM) signals. Here, a digital signal processor (DSP) is combined with the Matlab as the temperature controller for control the PWM signals. We control each block heating temperature by adjusting the duty cycle of the PWM values. The heating channel is Ch1 and the output channel is Ch4 in this control system. The transfer function of the temperature controlled object is identified as (21).

$$P(s) = \frac{2.854}{2395s+1} e^{-444.74s} \tag{21}$$



(a) Front view (b) Rear view

Fig. 7. Experimental setup of temperature control system

B. Simulation results

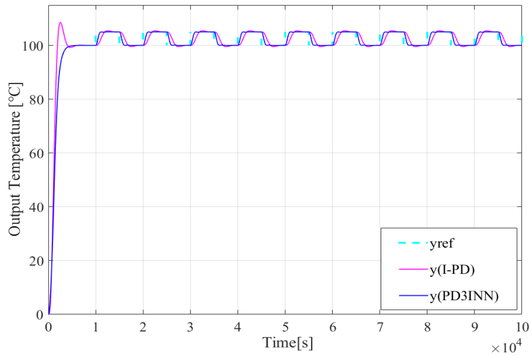
The reference model was setting as (22) based on the identified function parameters. Here, R is 0.01.

$$R(s) \approx \frac{2.854}{0.01*2395s+1} \times \frac{1}{(\frac{444.74}{2}s+1)^2} \tag{22}$$

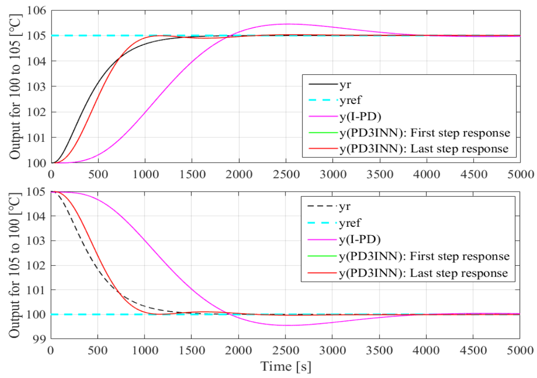
The parameters of I-PD controlled were K_p = 2.264, T_d = 222.37, and T_i = 889.48, respectively. And also, the initial values of those parameters in the RNN controller were α = 1×10^{-9} and β = 1×10^{-5}. The weights were initialized as an optimal (random) value between -1 and 1. The memory store steps z = 10, the hyper parameters of the *Adam* calculation were defaulted as β_1 = 0.99, β_2 = 0.99958, ε = 1×10^{-20}.

The simulation is divide into 2 parts. Part 1 was the reference value response without disturbance signal. Part 2 was the reference value response after adding disturbance. We separated each part into two steps. The first step is RNN control system learning period, the I-PD control plays a leading role at the initial stage(0-100 °C), after the RNN controller completes learning, the RNN controller mainly act to ensure quick and stable response for changes in the setting value. In the second step (RNN control period), a step signal of 5 °C is added into the input periodically. Hence, the temperature will go from 100 °C in a steady state to 105 °C and then follow the input signal return to 100 °C repeatedly. For convenience, temperature rise denotes the positive temperature control and temperature drop denotes negative temperature control. In Part 2, a 20% disturbance signal was added after the control system reached steady state in the control period. The RNN with PD compensator (named as PD3INN) and the RNN with 2DOF compensator (named as 2DOF3INN) were simulated. Compare the obtained time response performances with the results of classical I-PD control under the same conditions.

Fig. 8 (a) and (b) respectively show response curves at full time of the controller system, including positive and negative direction control results of the PD3INN system and I-PD control system. From the results, for the temperature positive control, the rise time of the temperature response in the PD3INN control and I-PD control is about 598.5 s and 1100s, respectively. The proposed method has an improvement of 46%. The settling time of the I-PD control and PD3INN is about 3587 s and 993.0 s, which has been improved by about 73%. Moreover, the overshoot of I-PD control is above 0.5 °C (10% of the setting value) is large, while the value of PD3INN control is zero. In the negative direction, compared with the drop time of the I-PD control is about 1100 s, the PD3INN control system is 598 s that has been improved by almost 46%. The settling time of I-PD control and PD3INN is about 3587 s and 993.0 s, PD3INN has an improvement of about 73%. Also, the I-PD control has an undershoot of 0.5 °C (10% of the setting value) while the value of PD3INN control is zero. In addition, compare the results of the last step time response with the first step response, results are almost the same. From these results, the RNN controller has completed learning, parameters are effectively updated and achieved good control performances. Although the figure shows a slight deviation at the initial stage, the actual controlled object can also follow the ideal reference model output quickly and steadily.



(a) Full response of PD3INN control system

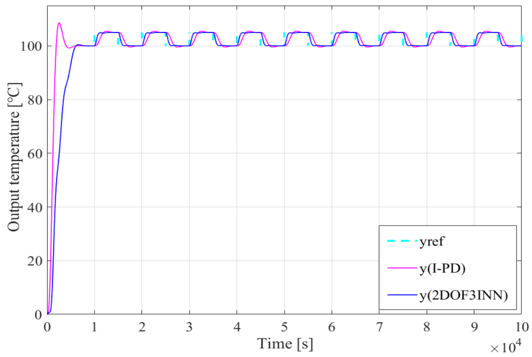


(b) Positive and negative direction control results

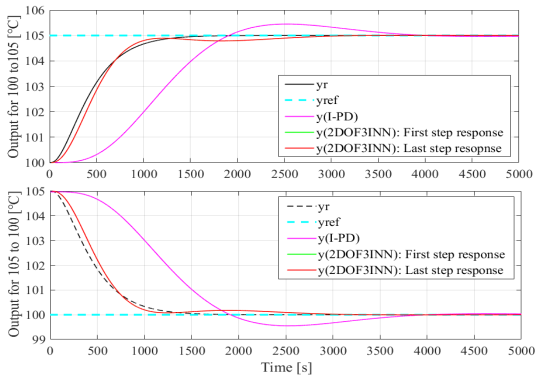
Fig. 8. Simulation results of PD3INN control system

Fig. 9 (a) and (b) respectively show the full time response and the two direction control results of the 2DOF3INN system. Compare response results of different control schemes and verify the proposed RNN method efficiency.

From the experiment results, the rise time (100°C-105°C) of the I-PD control and 2DOF3INN control is about 1101 s and 645.5 s, respectively. The 2DOF3INN has improved by about 41%. The settling time of the I-PD control system and 2DOF3INN control is about 3587 s and 2526.5 s, the proposed method result has been improved by about 30%. Also, the temperature overshoot is about 0.48°C in the I-PD control (10% of the setting value) while the result of 2DOF3INN is zero. For the negative direction temperature control, the drop time of the I-PD control and the 2DOF3INN control is about 1100 s and 645 s, respectively. The proposed method has an improvement of almost 41%. The settling time of I-PD control and 2DOF3INN control is 3587 s and 2526 s, has an improvement of about 30%. And also, the I-PD control has an undershoot of about 0.5 °C (10% of the setting value) while the proposed 2DOF3INN has no undershoot. As the same results of PD3INN control, the system response can successfully track the reference model. From these results, the proposed 2DOF3INN control can improve the control efficiency compared with the conventional control method. Compare the obtained experiment results of PD3INN control with those of 2DOF3INN control described as above, the performance of the PD3INN control is better than the 2DOF3INN control system.



(a) Full response for 2DOF3INN control system

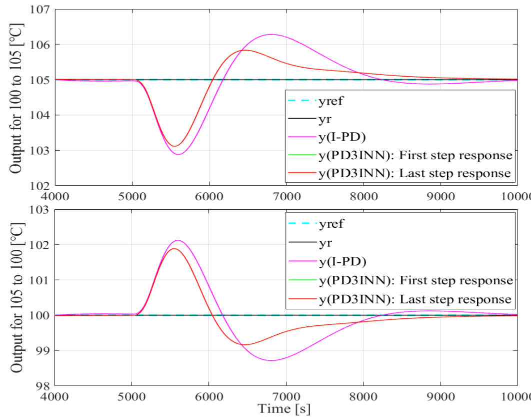


(b) Positive and negative direction control results

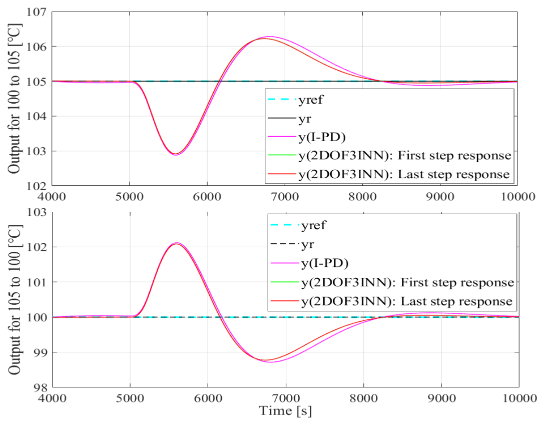
Fig. 9. Simulation results of 2DOF3INN control system

In Part 2, an amplitude of 20% disturbance was added to the control input after the system reached the steady-state in both directions. Fig. 10 (a) and (b) respectively show the disturbance response of PD3INN control and 2DOF3INN control in both direction control and the results of I-PD control were also used for comparison. The 20% disturbance is added at the time 5000 s in the positive direction, the similar analysis of the simulation results was carried out. The temperature drop of the I-PD control is 2.12 °C, while that of PD3INN and 2DOF3INN control systems are 1.9 °C and 2.08 °C, respectively. The temperature drop has been decreased with PD3INN and 2DOF3INN system by about 11%

and 2%, respectively. The settling time of the I-PD system (disturbance added) is about 4235.5 s, and that of PD3INN and 2DOF3INN control systems are 3500 s and 2998.5 s, respectively. The time was respectively improved by 17.4% and 29.2%. Moreover, the I-PD has an overshoot of 1.25 °C, while only 0.87 °C of PD3INN control system and 1.23 °C of 2DOF3INN control system. Similarly, after the disturbance was added in the temperature drop phase, the increased value in the I-PD control system is 2.12 °C, while that of PD3INN and 2DOF3INN control systems are 1.86 °C and 2.05 °C, respectively. Thus, the temperature drop has been decreased in PD3INN and 2DOF3INN systems by about 11% and 2%, respectively. The settling time of the I-PD system is about 4235 s, and time of PD3INN and 2DOF3INN control systems is 3498.5 s and 3000.5 s, respectively. The settling time was respectively improved by 17.4% and 29.2%. Moreover, the I-PD has an overshoot of 1.25 °C, while only 0.85 °C of PD3INN control system and 1.22 °C of 2DOF3INN control system, respectively. The response results of the both directions are almost the same, and both PD3INN and 2DOF3INN control systems have improved the control efficiency of the disturbance response.



(a) Disturbance response of PD3INN control system



(b) Disturbance response of 2DOF3INN control system

Fig. 10. Disturbance response simulation results

IV. Conclusion

This paper presents a RNN control method combined with a reference model that is proposed for improving the control performance of the time-delay temperature control system. The error between the ideal model output and the actual output drives the proposed RNN control system to achieve higher temperature control precision. The I-PD control output assists the RNN controller in the initial stage for stability control input of the controlled object. The proposed design method was applied to a real temperature plant. Simulations of both reference value response and disturbance response were carried out, respectively. From compared results above, the performance of system was greatly improved in rise time, settling time and overshoots. Meanwhile, the system has strong anti-interference ability in operation. The proposed RNN method can effectively the temperature response performance.

References

[1] K. Åström, "Pid controllers: Theory, design, and tuning," 1995.

[2] I. Nagrath, *Control systems engineering*. New Age International, 2006.

[3] A. Punse, S. Nangrani, and R. Jain, "A novel application of multipoint temperature control using pid," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019, pp. 1191–1196.

[4] J.-S. Ko, J.-H. Huh, and J.-C. Kim, "Improvement of temperature control performance of thermoelectric dehumidifier used industry 4.0 by the sf-pi controller," *Processes*, vol. 7, no. 2, p. 98, 2019.

[5] B. Song, J. K. Mills, Y. Liu, and C. Fan, "Nonlinear dynamic modeling and control of a small-scale helicopter," *International Journal of Control, Automation and Systems*, vol. 8, no. 3, pp. 534–543, 2010.

[6] M. Fliess, R. Marquez, and H. Mounier, "An extension of predictive control, pid regulators and smith predictors to some linear delay systems," *International Journal of Control*, vol. 75, no. 10, pp. 728–743, 2002.

[7] M. Xu and S. Li, "Practical generalized predictive control with decentralized identification approach to hvac systems," *Energy conversion and management*, vol. 48, no. 1, pp. 292–299, 2007.

[8] Z. Haiyang, S. Yu, L. Deyuan, and L. Hao, "Adaptive neural network pid controller design for temperature control in vacuum thermal tests," in *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 2016, pp. 458–463.

[9] V. Yadav, R. Padhi, and S. Balakrishnan, "Robust/optimal temperature profile control of a high-speed aerospace vehicle using neural networks," *IEEE transactions on neural networks*, vol. 18, no. 4, pp. 1115–1128, 2007.

[10] B. Dai, R. Chen, and R.-C. Chen, "Temperature control with fuzzy neural network," in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2017, pp. 452–455.

[11] K. Katić, R. Li, J. Verhaart, and W. Zeiler, "Neural network based predictive control of personalized heating systems," pp. 199–213, 2018.

[12] X. Li, T. Zhao, J. Zhang, and T. Chen, "Predication control for indoor temperature time-delay using elman neural network in variable air volume system," *Energy and Buildings*, vol. 154, pp. 545–552, 2017.

[13] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to sgd," *arXiv preprint arXiv:1712.07628*, 2017.

[14] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[15] S. Xu, S. Hashimoto, Y. Jiang, K. Izaki, T. Kihara, R. Ikeda, and W. Jiang, "A reference-model-based artificial neural network approach for a temperature control system," *Processes*, vol. 8, no. 1, p. 50, 2020.

[16] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, vol. 115, pp. 220–222, 1942.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.