Embedded Neural Network like PID Water Heating Controller Implementing Cycle by Cycle Power Control Scheme

Mustafa Q. Ali [1], *Karim* M. Aljebory [2,*], and *Monaf* S. Tapou [3]

[1]University of Baghdad, College of Islamic Sciences, Baghdad, Iraq.
[2]AL-Qalam University College, Technical Computer Engineering, Kirkuk, Iraq.
[3]AL-Qalam University College, Technical Computer Engineering, Kirkuk, Iraq.

Abstract. This paper experimentally investigates the heating process of a hot water supply using a neural network implementation of a self-tuning PID controller on a microcontroller system. The Particle Swarm Optimization (PSO) algorithm employed in system tuning proved very effective, as it is simple and fast optimization algorithm. The PSO method for the PID parameters is executed on the Matlab platform in order to put these parameters in the real-time digital PID controller, which was experimented with in a pilot study on a microcontroller platform. Instead of the traditional phase angle power control (PAPC) method, the Cycle by Cycle Power Control (CBCPC) method is implemented because it yields better power factor and eliminates harmonics in the power supply line. The smoothness of the heating process's output response, which is a result of both empirical experiments and simulation results, demonstrates the efficacy of the suggested control mechanism, where the output response had a small ripple margin. The system performed according to design expectations and had unimpaired unity power factor throughout its operating range and no ripple was detected during its functioning.

1 Introduction

Water heating is a very essential issue in domestic and industrial applications. In modern society, temperature control is not only used in industrial applications but also widely used in other fields, therefore it is of great significance to have means to measure and control temperature [1]. Numerous control applications implement the PID control scheme to maintain continual process performance. The proportional-integrative-derivative controller (PID) has an output that is a linear combination of its input, the integral of that input, and the derivative of the same input. Many classic tuning methods can be implemented to tune the gain parameters of a PID controller (K_p, K_i, K_d) such as the manual methods like trial and error method or rule based methods like Ziegler-Nichols and Kappa-Tau [2, 3, 4, 5, 6].

* Corresponding author: karim.eng@alqalam.edu.iq

Nevertheless, many industrial applications were often plagued with several complications like high-order constituents in their transfer function, time delays, and nonlinearities; for these reasons, it has been hard to tune the gains of their individual PID controllers. Therefore, a number of algorithms were developed that can tune these parameters more efficiently. There are several intelligent techniques that are compliant for applications on temperature control of water heating systems. These include different control schemes such as PID using Genetic Algorithms (GA-PID), Neural Networks (NN), Fuzzy Logic Control (FLC), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Genetic algorithm based Adaptive Neuro-Fuzzy Inference System (GA-ANFIS) [7]. In addition to those, there are additional techniques that work on the combination of fuzzy and PID control schemes as demonstrated in experimenting with a conventional PID controller in combination with fuzzy self-tuning techniques as described in reference [8] and [9]. Another researcher investigated the possible improvement of a heating process controller by applying model predictive control (MPC) strategies as seen in reference [10], the researchers implemented a nonlinear extended predictive self-adapting control algorithm to control the temperature of a heated water tank.

The approach proposed in this paper was implementing a digital neural network like self-tuning PID controller for the water heating process. This controller was tuned using the Particle Swarm Optimization (PSO) technique. Matlab was the platform used in developing a simulation model used for obtaining the tuning parameters required to ensure enhanced system response by implementing PSO tuning algorithm. These parameters are applied to the real time digital PID controller system on an ATmega328p microcontroller. This tactic does not require a combined structure of identification and decision as being implemented in standard self-tuning controllers.

The power control technique selected for this system was the Cycle by Cycle Power Control (CBCPC) scheme [11]. For an electric water heater, continuous electronic power control is preferable to conventional on-off control. Usually, to do this, the heating element—which is often a resistive load—is supplied with an AC mains voltage that is varied from its average value. The more common scheme of electronic power control in such applications is the phase angle power control (PAPC) method, which modifies the sinusoidal current waveform applied to the load by sculpting the rms voltage value of the load voltage using a semiconductor switching device, such as an appropriately rated TRIAC. This electronic switching element is placed between the ac supply and the controlled load. This power control scheme produces highly distorted current waveforms that produce high harmonic content that causes a number of issues in the electric supply grid and the other loads connected to it. Among the problems introduced are: Reduction of life time of transformers and motors due to overheating as a result of the harmonic introduced on the power line, increased power loss in conductors, polluting communication networks with electromagnetic interference, erroneous operation of circuit breakers, developing excessive currents in the neutral ac power line, degradation of Power factor because of the generated harmonic rms currents, and the addition of voltage notching to the waveform of the public ac supply. Furthermore, the public electric supply grid's waveform exhibits voltage notching due to this kind of ac power control method.

The CBCPC ac power control method implemented in this work does not introduce any harmonics to the ac power line compared to PAPC as it supplies the load with complete ac supply sinusoids separated by power cut intervals. As can be inferred from figure (1) where “m” cycles are applied to the load during the set time “T1” that has the minimum value of the period of one sinusoid which equal to 20mSec for 50 Hz rated mains supply chosen in the work done in this paper. “M” being the number of ac power cycles during the total time “T” chosen in accordance with the amount of resolution required by the control strategy of the process which is equal to 2 Sec in this work.

The power applied to the load is given by (1):

$$P_L = P_{IN} \frac{m}{M} \tag{1}$$

(P_{IN}) is the input power, (P_L) is the power delivered to the load. The ratio "m/M" given by formula (1) depends on the required precision for the designed controller, which is selected in this work to vary between 0.01 and 1. Which means supplying 1% to 100% power to the load according to controller demand.

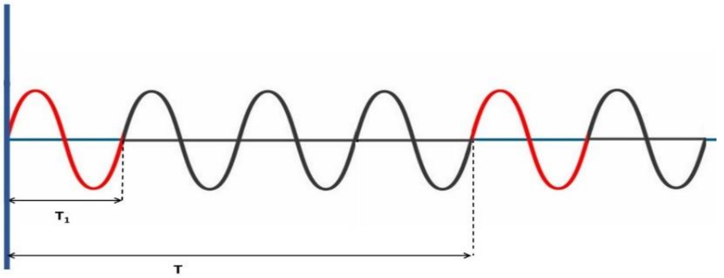

Fig. 1. Cycle-by-cycle (integral-cycle) power control scheme.

The remaining sections of this work are arranged as follows: The second section outlines the suggested control method, describes the water heating system's designed real-time temperature control system, and analyzes the hardware and mathematical model of the system. The suggested neural network-like self-tuning PID controller and its corresponding tuning algorithm are deduced in Section Three. The results of the simulation are shown in Section Four, along with an application of the suggested control strategy to real-time computer control on the ATmega328p microcontroller-based platform. Section Five provides the conclusions.

2 The Experimental Work

The experimental work done in this work involves the design and implementation of the real time water temperate controller hardware and software. In these experiments, a standard IBM compatible laptop working under Windows10 operating system is the platform for running the simulations under Matlab and recording real time results. Figure (2_a) shows a schematic of the system for controlling water temperature, while figure (2_b) shows the experimental setup. The setup consists of a water heating element, a water tank, and a one wire communication digital temperature sensor DS18B20. The Mains ac power source supplies power to the water heating element through the (BTA41) TRIAC. The ATmega328p microcontroller running at 16MHz is incorporated in the control circuit. It receives input from temperature sensor and generates the appropriate controlling signal to the TRIAC in accordance with the CBCPC power control scheme to deliver the required power magnitude to the heating element in the water heating process. This process is characterized as a first order time delay system in the mathematical model. The heating element has a 1.5KW rating. It generates heating power in the range 0 KW to 1.5 KW as heating process input and the output is water temperature inside the tank.

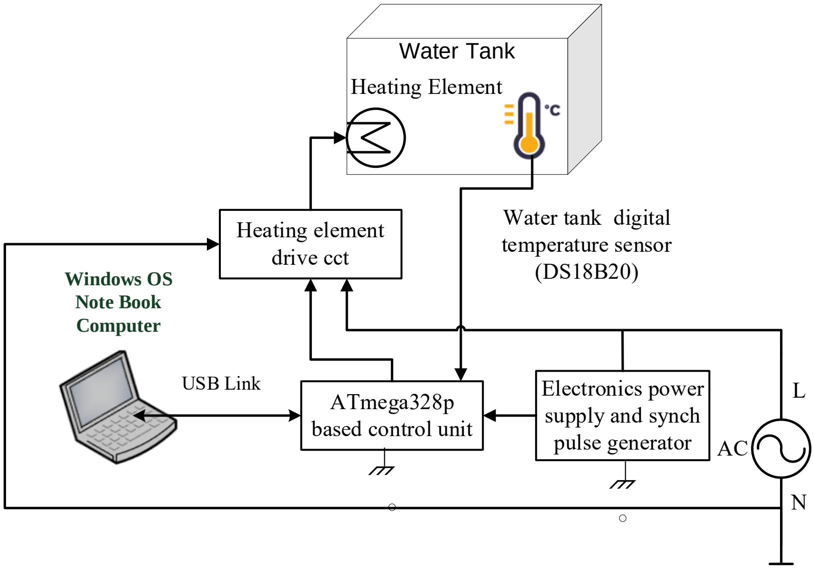

Fig. 2-a. Schematic of the system for controlling water-temperature.

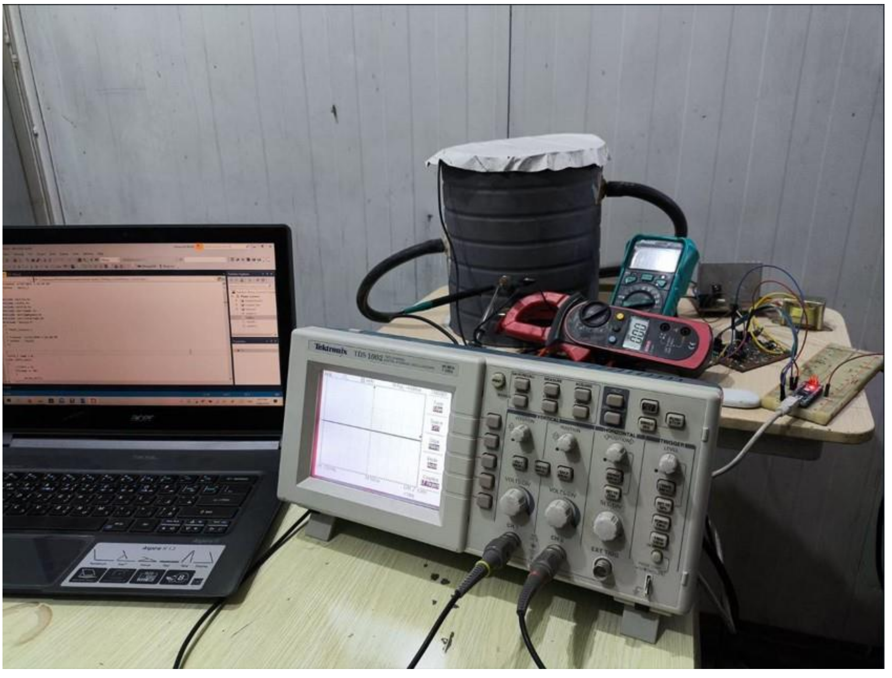

Fig. 2-b. The layout of experimental setup.

2.1 System's Numerical Model

The heat balance equation can be used to drive the heating system numerical model in the following way:
The input heat – The output heat = The Build-up Heat

$$Q_{input}(t) - Q_{output}(t) = Q_{build-up}(t) \tag{2}$$

The Unsteady-state of the build-up heat equation is:

$$Q_{input}(t) - HA\left(T\ (t) - T_{air}(t)\right) = M_{CP}\frac{d\ T\ (t)}{d\ t} \tag{3}$$

The Steady-state of this equation when t is equal to 0:

$$Q_{input}\ (0) - HA\left(T\ (0) - T_{air}(0)\right) = 0 \tag{4}$$

The disparity between the two equations is shown as:

$$Q_{input}(t) - HA\ T\ (t) = M_{CP}\frac{d\ T\ (t)}{d\ t} \tag{5}$$

The Laplace Transform of the above equation is:

$$Q_{input}(s) - HA\ T\ (s) = M_{CP}\ ST\ (s) \tag{6}$$

$$\frac{T\ (s)}{Q_{input}(s)} = \frac{\frac{1}{HA}}{\frac{M_{CP}}{HAS + 1}} \tag{7}$$

$$\frac{T\ (s)}{Q_{input}(s)} = \frac{K}{\tau\, S + 1} \tag{8}$$

The value of $K = \frac{1}{H\ A}$ and $\tau = \frac{M_{CP}}{H\ A}$.

The actual parameters of the water heating process under investigation are given in table (1).

Table 1. The actual parameters of the investigated water heating process.

Actual parameters	**Values**
H: The coefficient of heat transfer	11 $watt / m^2C^o$
Cp: the heat of water	4.2 KJ / KgC^o
M: the mass of the water	7.48 Kg
A: the tank's surface area	0.075 m^2

Equation (9) depicts the equation describing the heating system numerical model after these parameters are applied to equation (7):

$$\frac{T\ (s)}{Q\ input\ (s)} = \frac{1.22}{10.2\,S + 1} \tag{9}$$

2.2 System Hardware Architecture

Figure (3) shows the water heating control circuit's architecture. It includes the DS18B20 digital temperature-sensing element and one-wire communication protocol. The output of this digital sensor is connected to the input port pin PD3. The next section of the hardware is the electronics power supply and synch pulse generator [12, 13]. This section provides the required dc source to operate the system electronics and generates the synch pulses that are fed to the ATmega328p microcontroller counter/timer module TC0, the input to this counter comes from the output of the AC synch pulse generator. The USB to RS232 TTL module (PL2303HX) is employed to provide communication between the laptop computer and the ATmega 328p microcontroller. Electrical isolation between the gate of the load driving TRIAC device BTA41 and the control output from the microcontroller is provided through MOC3021 TRIAC opto-isolator to control the AC power fed to the 1.5KW heating element. This opto-isolator guarantees that the Power control TRIAC (BTA41) is always turned on at the zero crossing point of the ac power line voltage.

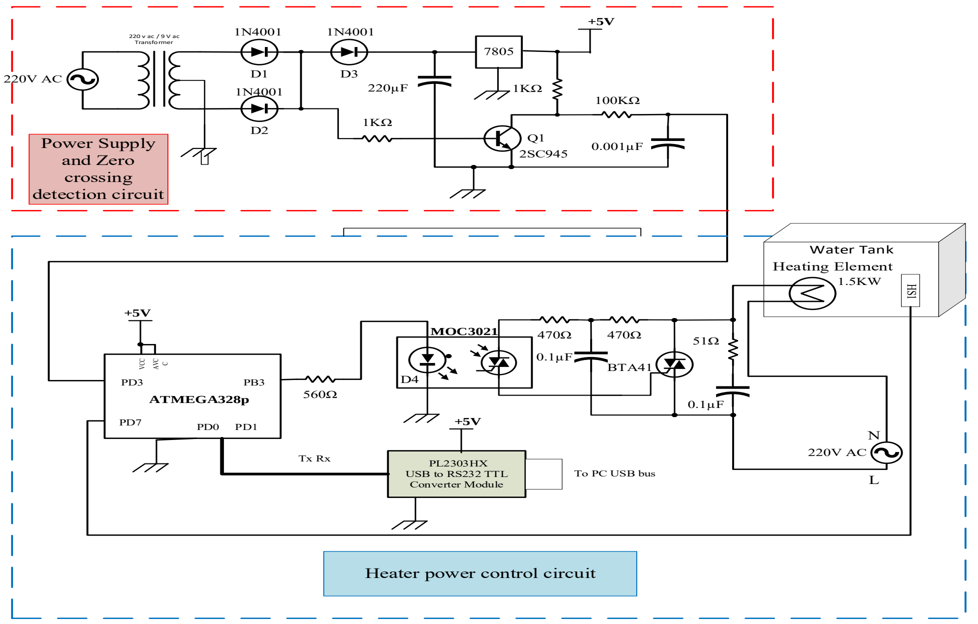

Fig. 3. Schematic of the circuit for controlling water-temperature.

3 PID Neural Controller Approach

The water temperature inside the tank can be controlled by varying the electric power delivered to the heating elements. The typical block diagram of a PID neural controller is shown in figure (4).

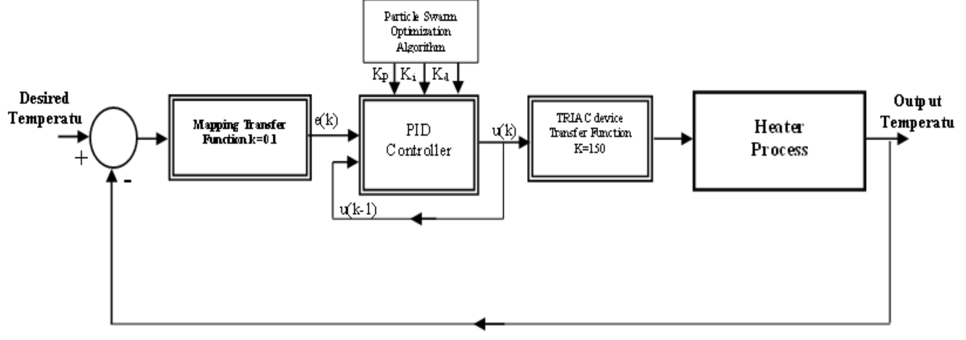

Fig. 4. The schematic representation for the heating process controller illustrates how the simulation run in the Matlab environment is mapped to the real-time process.

The primary duties of the PID neural controller are to minimize tracking error of the system output when it deviates from the desired set value and to maintain system stability.

Figure (5) illustrates the topology of the self-tuning PID neural controller that has been implemented. It has a structure based on a typical PID controller arrangement with three terms: proportional, integral, and derivative. A PID controller has the numerical form given s-domain as shown in equation (10) [14].

$$Gc\ (s) = P + I + D = K_p + \frac{K_i}{S} + K_d\ S \qquad (10)$$

In formula (10), K_p is the proportional component, K_i is the integral component, and K_d is the derivative component.

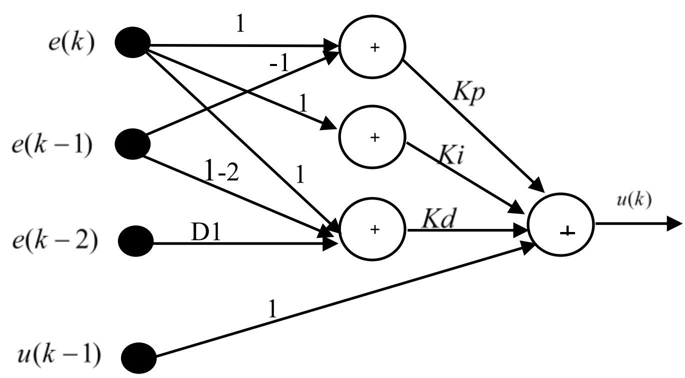

Fig. 5. The architecture of a self-tuning neural network like PID controller.

The PSO algorithm technique is used in adaptive self-tuning to modify the PID neural controller's parameters. The suggested self-tuning PID neural control technique is similar to the discrete-time equation (11)'s neural network PID controller structure. [15].

$$u(k') = u(k' - 1) + Kp[e(k') - e(k' - 1)] + Kie(k') + Kd[e(k') - 2e(k' - 1) + e(k' - 2)] \qquad (11)$$

As a result, e (k'), e(k'-1), e(k'-2), and u(k'-1) are the self-tuning PID input vector's components, where e (k') stands for the input error signals and u(k'), u(k'-1) for the self-tuning PID output.

3.1 The Learning Algorithm

The learning algorithm chosen for this work is called Particle Swarm Optimization (PSO). It mimics the movement of flying bird flocks while finding the best possible solution. The PSO algorithm uses a population of individuals that are called "particles", these particles *fly* over the solution space in search of the best solution.
Every particle moves across the search space at a different place and speed. A fitness function is used to assess these particles in order to determine how near the ideal solution they are. [16, 17, and 18].
The preceding best value is designated "*pbest*". This *pbest* value is associated with one distinct particle. There is another value called "*gbest*", it corresponds to the best value given for the group of particles *pbest* in the swarm.
To generate a particle, the neural network is rebuilt as an array, similar to a self-tuning PID controller with three weight parameters of the PID controller matrix. These Particles are

given initial values arbitrarily, and then are updated afterward in accordance with equations (12, 13, 14, 15, 16, 17) [16, 17, 18] so as to find the best tuning parameters for the given PID controller:

$$\Delta Kp_m^{k'+1} = \Delta Kp_m^{k'} + c_1 r_1 \left(pbest_m^{k'} - Kp_m^{k'}\right) + c_2 r_2 \left(gbest^{k'} - Kp_m^{k'}\right) \tag{12}$$

$$Kp_m^{k'+1} = Kp_m^{k'} + \Delta Kp_m^{k'+1} \tag{13}$$

$$\Delta Ki_m^{k'+1} = \Delta Ki_m^{k'} + c_1 r_1 \left(pbest_m^{k'} - Ki_m^{k'}\right) + c_2 r_2 \left(gbest^{k'} - Ki_m^{k'}\right) \tag{14}$$

$$Ki_m^{k'+1} = Ki_m^{k'} + \Delta Ki_m^{k'+1} \tag{15}$$

$$\Delta Kd_m^{k'+1} = \Delta Kd_m^{k'} + c_1 r_1 \left(pbest_m^{k'} - Kd_m^{k'}\right) + c_2 r_2 \left(gbest^{k'} - Kd_m^{k'}\right) \tag{16}$$

$$Kd_m^{k'+1} = Kd_m^{k'} + \Delta Kd_m^{k'+1} \tag{17}$$

$m = 1,2,3,\ldots\ldots number\ of\ particles.$

$k_m^{k'}$ is the phrase used to describe particle m's weight at k' iteration..
The acceleration constants with positive values of (2) are denoted by the letters c1, and c2.
The two random numbers, r1 and r2, have values between 0 and 1.
$pbest_m$ is the m^th particle's best prior weight.
$gbest$ refers to the highest particle value in the group of all the particles.
The PID controller has three parameters therefore in this work the number of dimensions in particle swarm optimization will be equal to three accordingly.
The mean square error function is selected as the criterion chosen for estimating the proposed model performance is the mean square error function that is calculated from equation (18):

$$E = \frac{1}{2}\sum_{j=1}^{pop}(T_{ref}\ (K+1)^j - T_{out}(k+1)^j)^2 \tag{18}$$

The following procedures are carried out in order to generate PSO for the neural network-like self-tuning PID controller:

- **Step1** Establish the starting points of the search randomly. Kp_1^0, Ki_1^0, Kd_1^0, ΔKp_1^0, ΔKi_1^0 and ΔKd_1^0 of each particle within the allowable range. It should be noted that all of the neural network's parameters, such as the PID controller shown in figure (5), are included in the search space dimension. Each particle current search point is assigned the value of *pbest*. The *gbest* weight is set to the best estimated value of *pbest* and the particle number having the finest value is stored.
- **Step2** From equation (12) the objective function value is calculated for every particle in the swarm. This new value is to *pbest* if it is better than the current value of pbest for a particular particle. On the other hand, *gbest* is substituted with the best value, and the particle number containing the best value is stored if the best value of *pbest* is greater than the current value of *gbest*.
- **Step3** In this step, equations are applied to update the current search position for every particle in the swarm. (12, 13, 14, 15, 16, and 17).
- **Step4** The optimization process has ended and the program is terminating if the current iteration count has reached the maximum number that has been set. If not, the process moves on to step 2.

4 Simulation and Experimental Results

In this section, the mapping between the real process and the simulation Matlab package is applied as shown in Figure (5). The heater unit is suggested to be used with the TRIAC device transfer function, which is linear with a saturation transfer function that has a gain of 150 and a maximum output power of 1500 watts. This figure is dependent on the heating element ratings.

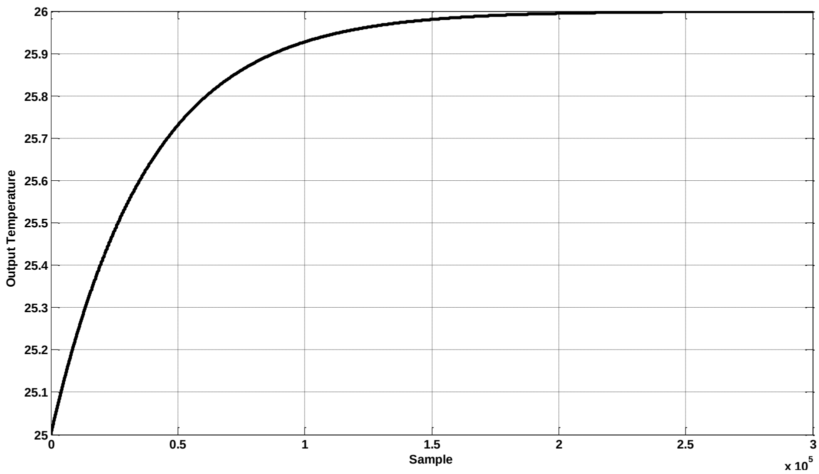

Fig. 6. Step response of open loop system.

Figure (6) displays the open loop response of the water heating process for a step input. When applying a step change in the power utilized by the heating element equaling to (21.45) watts, this will cause a rise in water tank temperature equaling one degree Celsius which corresponds to a rise in tank water temperature from 25 º*C* to 26 º*C*. This open loop response is used to find the time constant of the process under investigation and it was found being equal to 36720 seconds. Matlab platform is used to perform the simulation for the water heating process. The PSO learning algorithm steps are carried out to obtain the best tuning parameters for the neural network like the self-tuning PID controller of the system shown in the block diagram of figure (4), a sampling interval (T_S) of 1sec is selected. The PSO tuning algorithm parameters are set to the following values:

- The particle population is set to 25.
- The number of iterations is set to 100.
- The number of weights assigned to each particle is set to 3 as there are three parameters in the PID controller.

Then using four steps of the learning algorithm running through 100 iterations and applying the resulting tuning parameters to the neural network like PID controller in Matlab simulation, the resulting response figures are illustrated in figures (7) respectively for the initial temperature condition of 25 º*C*. The process response shown in Figure (7-a) took 540 seconds to reach a steady state. A small oscillation in control action at the steady state is observed. This is due to the nature of the closed-loop controllers as shown in Figure (7-b). The gains of the PID self-tuning neural controller are shown in Table (2), then these parameters are used in the practical experimentation.

Table 2. The best PID controller parameter by using PSO algorithm.

Kp	Ki	Kd
103.192	0.125	0.393

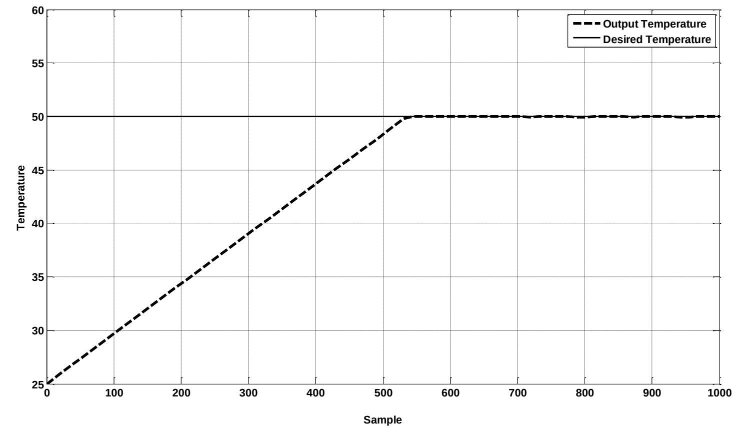

Fig. 7-a. The heater process temperature output.

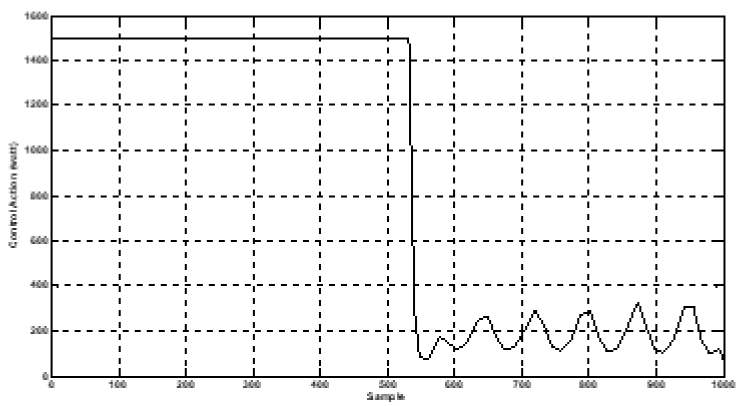

Fig. 7-b. The control Action.

Figure (8) shows the real-time response of the water temperature control system. The initial temperature condition was 25º*C* and from the results of the control action, larger oscillations in steady-state temperature were observed than those obtained in the simulation especially at steady state set point of 50º*C*. The steady-state error in real time was equal to approximately ±1º*C*, but in carried out simulations, it was equal to zero. The offset of the instrumentation amplifier, the quantization error of the analog-to-digital converter of the digital temperature sensor, and other undesired features of the implemented temperature sensor, such as non-linearity, drift, and offset, are the causes of this real-time steady-state error. The oscillation seen in the system response is caused by these causes.

While the CBCPC power control method was usually associated with the flicker effect, it was only noticeable while using classic incandescent lamps for lighting, no flicker was observed while using modern compact fluorescent lamps or LED lamps for lighting.

The power factor was always unity and no harmonics were generated while applying the CBCPC scheme.

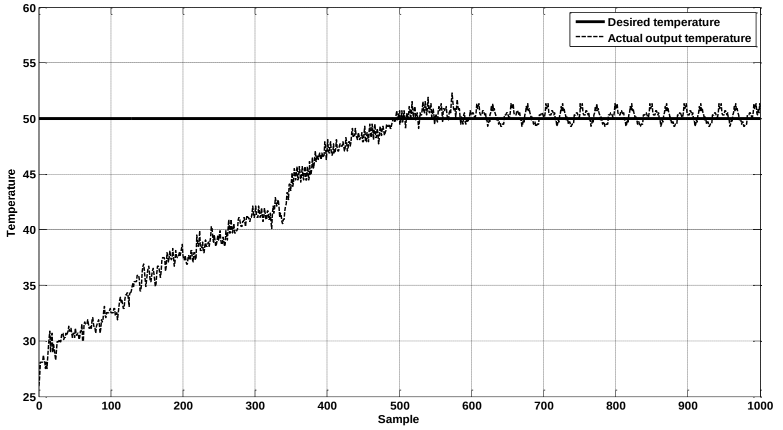

Fig. 8-a. The actual output temperature.

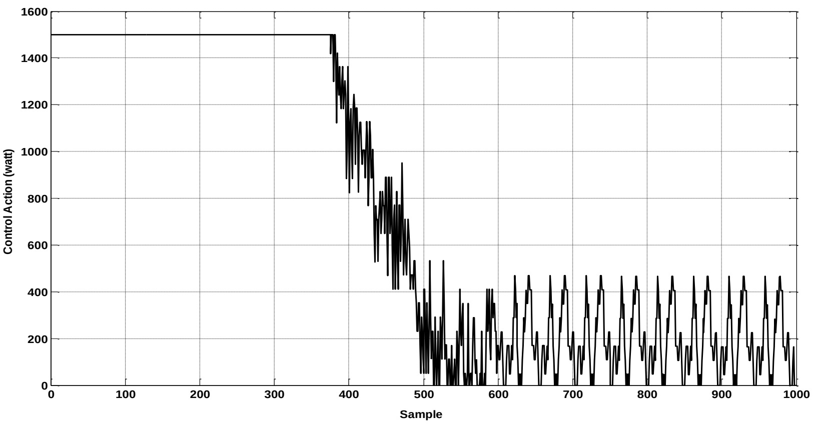

Fig. 8-b. The actual control action.

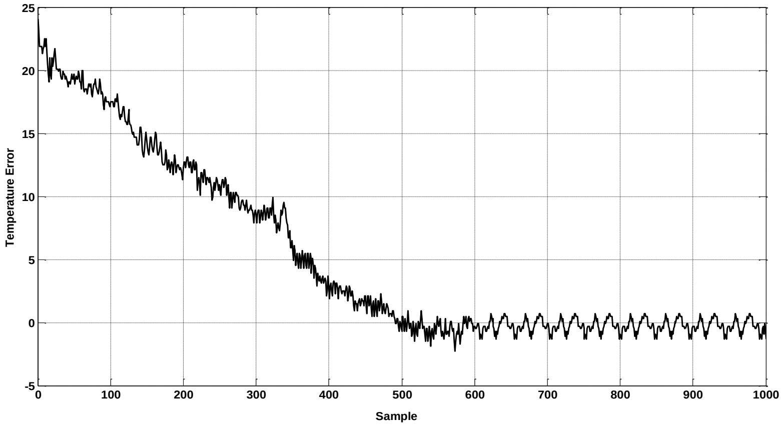

Fig. 8-c. The actual temperature error.

5 Conclusion

In this paper, the experimental work in real-time computer control of water temperature in a tank using CBCPC power control scheme and a neural network-like digital controller implemented in an ATmega328p microcontroller with the DS18B20 digital temperature sensor was investigated.

The optimal system tuning parameters for the neural network, such as a self-tuning PID controller, were discovered using the particle swarm optimization algorithm technique. The PID controller's tuning parameters were then applied to the discrete-time real-time computer control. The real-time controller implemented using the ATmega328p platform using the CBCPC power control scheme resulted in a water temperature control system that maintained water temperature in a tank with less than ±1ºC precision. No flicker effect was observed due to the implementation CFL and LED lamps instead of the classic incandescent lamps in modern lighting systems. The water heating system power factor was always unity throughout the power operating range from 0% to 100% and no harmonics were observed on the source supply waveform.

This work can be further enhanced by introducing additional parameters to the controller input, these include introducing an additional temperature sensor that monitors inlet water temperature and a flowmeter to measure the amount of heated water consumed. This will enhance estimating the energy required to keep tank water temperature as close as possible to the desired set point.

References

1. W-T. Huang and J-P. LI, Research and design of intelligent temperature control system. Proceeding of 2010 Second International Workshop on Education Technology and Computer Science, 2010, pp. 538-541.
2. T. Hagglund, K. Astrom, Automatic tuning of PID controllers, in W.S. Levine (Ed), the Control Handbook, CRC Press, Boca Raton, FL, 1996, pp. 817-826.
3. Muresan, C. I., & De Keyser, R. (2022). Revisiting Ziegler–Nichols. A fractional order approach. *ISA transactions*, *129*, 287-296.
4. Zhu, J., Yang, Q., Lu, J., Zheng, B., & Yan, C. (2015). An adaptive artificial neural network-based supply air temperature controller for air handling unit. *Transactions of the Institute of Measurement and Control*, *37*(9), 1118-1126.
5. S. Muhammed, Internal model control structure using adaptive inverse control strategy. ISA Transaction vol. 44, 2005, pp. 353-362.
6. Mikhalevich, S. S., Baydali, S. A., & Manenti, F. (2015). Development of a tunable method for PID controllers to achieve the desired phase margin. *Journal of Process Control*, *25*, 28-34.
7. S. Saini and S. Rani, Temperature Control Using Intelligent Techniques. Proceeding of 2012 Second International Conference on Advanced Computing & Communication Technologies, 2012, pp. 138-145.
8. Z. Dong, Y. Su and X. Yan, Temperature Control System of the Thermal Analyzer Based on Fuzzy PID Controller. Proceeding of 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 58-61.
9. J. Wei, Research on the Temperature Control System Based on Fuzzy Self-tuning PID, 2010 International Conference On Computer Design And Applications (ICCDA 2010), pp. 13-15.

10. M. Sbarciog, R. Keyser, S. Cristea and C. Prada, Nonlinear predictive control of processes with variable time delay. A temperature control case study. Proceeding of 17th IEEE International Conference on Control Applications Part of 2008 IEEE Multi-conference on Systems and Control San Antonio, Texas, USA, September 3-5, 2008, pp. 1001-1006.
11. C.L. Lorenzetti. "Direct Water Heater Power Control for Reduced Harmonics and Flicker Content with Optimized Half-cycle Power Control". Int. Revista da Propriedade Industrial, Vol 11, issue 3,PP. 175—180, Seção I. Rio de Janeiro, Brazil, November 2006.
12. C.J. Savant, S. Roden and L. Gordan, Electronic design: circuits and systems, Benjamin/Cummings Publishing Company Inc. 1991.
13. Miguel Castilla, Control Circuits in Power Electronics, The institution of engineering and Technology. London, United Kingdom. 2016.
14. Q. Zhong, Robust control of time-delay systems, Springer – Verlag London Limited 2006.
15. Halici, U., Leblebicioglu, K., Özgen, C., & Tuncay, S. (2018). Recent advances in neural network applications in process control. *Recent Advances in Artificial Neural Networks*, 229-289.
16. Carrasco, J., García, S., Rueda, M. M., Das, S., & Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation*, *54*, 100665.
17. Wang, C. C., Kuo, P. H., & Chen, G. Y. (2022). Machine learning prediction of turning precision using optimized xgboost model. *Applied Sciences*, *12*(15), 7739.
18. Yasuoka, Y., Shinomiya, Y., & Hoshino, Y. (2016, August). Evaluation of optimization methods for neural network. In *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)* (pp. 92-96). IEEE.