IMPROVE SYSTEM STABILITY USING NEURAL HYBRID CONTROLLER-PLC

Thanh Son Huynh, Hong Ngan Vo*

Dong Nai Technology University

*Corresponding author: *Hong Ngan Vo, vohongngan@dntu.edu.vn*

GENERAL INFORMATION

Received date: 09/03/2024

Revised date: 08/05/2024

Accepted date: 11/07/2024

KEYWORD

Neural;

PLC;

PID;

RBF;

SCL

ABSTRACT

This paper presents methods for controlling a real model using the S7-400 controller with SCL language (Structured Control Language). The two controllers designed are the hybrid Neural Network Neural-PID (Proportional Integral Derivative) controller and the RBF (Radial Basis Function)-PID controller. The control results of the real model, a single water tank, give quite good results, the errors in all cases are small, and the overshoot is small. In the two cases above, the system operates stably and the Neural-PID hybrid controller gives the best results. The hybrid Neural-PID controller has both the stability of a PID controller and the adaptive learning of a Neural controller. Therefore, this method is capable of controlling other models in industry such as controlling weighing conveyors in the cement industry, controlling heating systems, etc.

1. INTRODUCTION

With the continuous development of society, the application of scientific research results to industrial production has become extremely important. Neural networks are not a new topic and there are many studies and applications in this field. However, the development of a Neural-PLC (Programmable Logic Controller) controller for industry is still an essential need in the context of rapidly developing industry in Vietnam. Currently, Neural-PLC controllers used in some factories in Vietnam are mainly copyrighted from the manufacturer. Therefore, the goal of the study is to apply knowledge of Neural networks combined with PLC to design a Neural-PLC controller in industry (Topalova & Tzokev, 2010); Ahmad & Prajitno, 2020). Traditional PID controllers are notable for their simple structure, easy adjustment, low cost and effective response ability (Combaluzier et al., 2016; Coelho et al., 2020). Meanwhile, artificial neural networks can be considered as a basic mathematical model of the brain, operating as a distributed computing network (Golenkov et al., 1992). Unlike traditional computers, which need to be programmed to perform specific tasks, most neural networks require training (Wu & Feng, 2018). They are capable of learning new connections, functional relationships, and new patterns. Neural networks are a fundamental tool for developing intelligent systems that can learn. One of the outstanding advantages of neural networks is their adaptability. Thanks to the ability to automatically adjust weights, neural networks can optimize operations such as pattern recognition and system control decision making. This adaptability allows neural networks to maintain operating efficiency when the environment and control objects change over time (Grossberg, 2013).

Based on his own practical needs, the author chose the research topic of building a Neural controller in industry by building functions in Siemens controllers. Using the SCL language available in SIMATIC (Berger, 2012), the author studies the algorithms of three controllers: PID controller (Al Gizi et al., 2015), Neural controller (Alber et al., 2019), RBF controller (Arora et al., 2014). Each controller has different advantages and disadvantages. To create different control purposes, there are combinations of advantages between PID and Neural network into the hybrid control method Neural-PID (Webb et al., 2011) and similar to the RBF-PID controller (Ma et al., 2020). In this article, the author will present two main controllers, the Neural-PID controller and the RBF-PID controller to control a real model of a single water tank. The measurement results show that both controllers operate stably, but the hybrid Neural-PID controller gives the best response control results, the errors in all cases are small, and the overshoot is small.

2. OBJECTS AND METHODS

2.1. Research object

The control object is a single water tank model as shown in Figure 1. A single water tank is a nonlinear object, the water level signal is often fluctuating (with interference), so it is easy to simulate. Simatic manager programming software is specialized for S7-300 and S7-400 controllers, combined with Omron's E4PA Ultrasonic Sensor, MM420 inverter and Redlion's industrial screen (Touch Screen) to control the speed of the pump into the water tank.

Using the SCL language available in Simatic, the author researches two main controllers: the Neural-PID controller and the RBF-PID controller. Combining the advantages of PID and Neural into a hybrid Neural-PID control method and combining RBF and PID into a hybrid RBF-PID control method. The topic uses artificial neural network technique combined with traditional PID controller to design a controller with good output response.

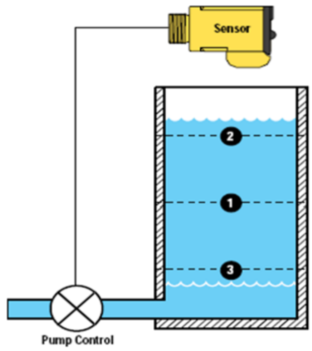


Figure 1. Water tank control model

2.2. Research method

The topic uses the experimental method to find the optimal controller for the object here is a single water tank. The author studies two main controllers including the Neural-PID controller and the RBF-PID controller. The goal is that the controller must be stable, sustainable, optimal, at the same time the cost of use must be low and must be widely applied in life. Experimental results show that the hybrid Neural-PID controller controlling a single water tank gives quite good results, with small errors.

3. SYSTEM DESIGN

3.1. Single water tank model

Consider a water tank system containing liquid with a cross-section that varies with height as shown in Figure 2 as follows:

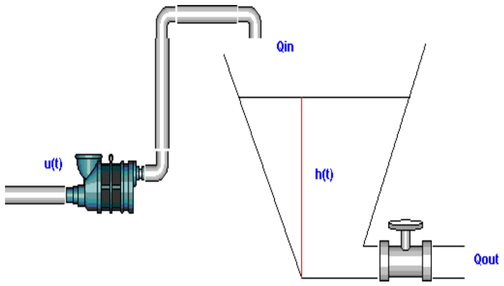


Figure 2. Single Tank Model

The differential equation describing the system is:

$$\dot{h}(t) = \frac{1}{A(h)}\left(ku(t) - C_D a\sqrt{2gh(t)}\right) \quad (1)$$

$$A(h) = \frac{A_{max} - A_{min}}{h_{max}} h + A_{min} \quad (2)$$

In which:

$u(t)$ – pump control voltage ($0 \leqslant u(t) \leqslant 10$V)

$h(t)$ – liquid level height in the tank (cm)

$A(h)$ – cross-sectional area of the tank (cm^2)

h_{max} – maximum height of the tank

A_{max}, A_{min} – maximum and minimum cross-sectional areas

k – coefficient proportional to pump capacity

a – discharge valve cross-sectional area (cm^2)

g – gravitational acceleration (981cm/sec^2)

C_D – discharge coefficient

Parameters of single water tank system are as Table 1:

Table 1. Parameters of single water tank system

Parameter	Value
Maximum height, h_{max} (cm)	50
Maximum cross section, A_{max} (cm^2)	100
Minimum cross section, A_{min} (cm^2)	1
Proportional coefficient to pump capacity, k (cm^3/sec)	300
Exhaust valve cross section, a (cm^2)	1
Discharge coefficient, C_D	0.6

The problem is to control the liquid level in the tank according to the set signal. From the differential equation (1) above we have:

$$\frac{h(t+1) - h(t)}{\Delta t} = \frac{1}{A(h)}\left(ku(t) - C_D a\sqrt{2gh(t+1)}\right) \quad (3)$$

Subtracting $h(t+1)$ we get:

$$h(t+1) = \frac{\Delta t}{A(h)}\left(ku(t) - C_D a\sqrt{2gh(t+1)}\right) + h(t) \quad (4)$$

Substituting the value of $A(h)$ from equation (2) we have:

$$h(t+1) = \frac{\Delta t}{\frac{A_{max} - A_{min}}{h_{max}} h(t+1) + A_{min}} \cdot \left(ku(t) - C_D a\sqrt{2gh(t+1)}\right) + h(t) \quad (5)$$

Substituting the values of A_{max}, A_{min}, h_{max}, k, C_D, a, g, with sampling period t = 1s into equation (5), we have:

$$h(t+1) = \frac{1}{1.98h(t+1) + 1} \cdot \left(300u(t) - 26.58\sqrt{h(t+1)}\right) + h(t) \quad (6)$$

The single tank system is a nonlinear system. In the simulation run, we will put this mathematical equation into the object section. When running with the real model, the object will be the actual external tank. Here we have one output that needs to be controlled, $h(t+1)$ and two input variables, $u(t)$ and $h(t)$.

3.2 Closed-loop numerical control system:

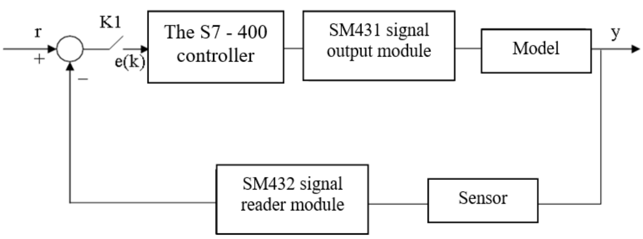
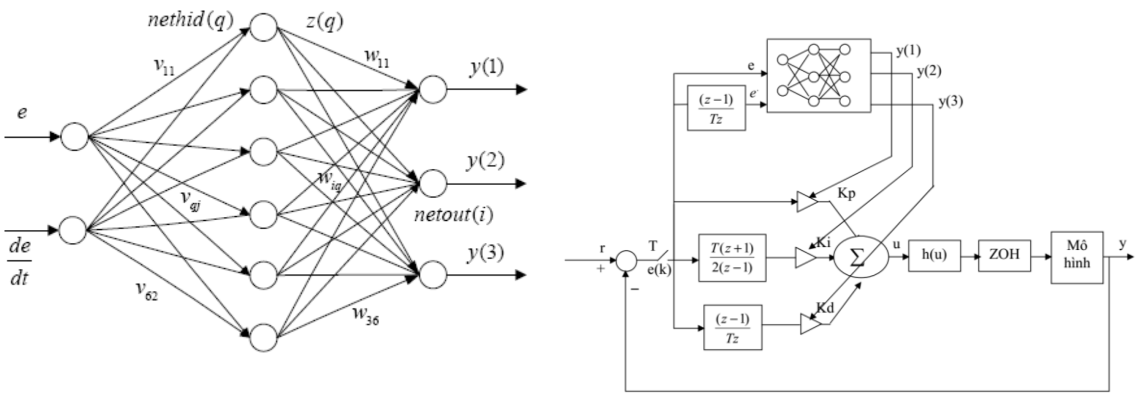


Figure 3. Closed-loop digital control system

The closed-loop digital control system is shown in Figure 3. Through the *K* (symbolic) keys working synchronously, the S7-400 controller reads the feedback signal from the SM431 Module that has received the signal from the water level sensor, compares it with the signal set from the industrial screen, then processes it according to two algorithms Neural-PID, RBF-PID and outputs the control signal. The control signal is transferred to the SM432 analog signal output module to control the object.

3.2.1 Neural-PID controller



(a) (b)

Figure 4. Neural-PID controller: (a) Neural connection diagram, (b) Neural-PID control system

The Neural-PID controller has the stability of the PID controller and the adaptive learning of the Neural network (Yu & Rosen, 2013; Rossomando & Soria, 2015). The Neural-PID controller is a 3-layer feed-forward network: an input layer, a hidden layer, and an output layer. The input layer has 2 neurons that receive signals from *e* and *dt*/*de*. The hidden layer has 6 neurons, using a bipolar S-shaped activation function. The output layer has 3 neurons, using a Sigmoid activation function. The three outputs are responsible for adjusting the 3 parameters K_P, K_I, K_D respectively. All neurons are connected together as shown in Figure 4(a), 4(b).

$x_j\ (j = \overline{1,m})$: the input vector consists of two neurons that receive signals from *e* and *de*/*dt*.

$z_q\ (q = \overline{1,l})$: output of the *q* unit of the hidden layer with *l* equal to 6.

$y_i\ (i = \overline{1,n})$: are the outputs of the output layer with *n* equal to 3.

$v_{qj}\ (q = \overline{1,l}; j = \overline{1,m})$: is the connection weight between the input layer and the hidden layer.

$w_{iq}\ (i = \overline{1,n}; q = \overline{1,l})$ is the connection weight between the hidden layer and the output layer.

The weighted sum of the inputs to the qth neuron in the hidden layer is:

$$net_q = \sum_{j=1}^{m} v_{qj} x_j \quad (7)$$

The output signal of the qth neuron in the hidden layer is:

$$z_q = a_h(net_q) = a_h\left(\sum_{j=1}^{m} v_{qj} x_j\right) \quad (8)$$

The weighted sum of the input signals to the ith neuron in the output layer is:

$$net_i = \sum_{q=1}^{l} w_{iq} z_q = \sum_{q=1}^{l} w_{iq} a_h(net_q) = \sum_{q=1}^{l} w_{iq} a_h\left(\sum_{j=1}^{m} v_{qj} x_j\right) \quad (9)$$

The output signal of the ith neuron in the output layer is:

$$y_i = a_0(net_i) = a_0\left(\sum_{q=1}^{l} w_{iq} z_q\right) \quad (10)$$

Suppose we have a training dataset consisting of K samples $(x(k), d(k))$, $k = \overline{1,K}$

The criterion for training the network is to minimize the error:

$$E = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2 \quad (11)$$

The function E forms a convex surface in space, on which there is a minimum point. With any set of weights, we can calculate the value of E on the convex surface.

The program of the Neural-PID controller as shown in Figure 4 (b) is written using the FC3 function in the SCL language. After compiling the SCL program, the FC3 function will be created in the Block of the program according to the functions from (12) to (17).

Hidden layer activation function:

$$f_{hid} = \frac{2}{1 + e^{-0.3net}} - 1 \quad (12)$$

Output layer activation function:

$$f_{out} = \frac{2}{1 + e^{-0.1net}} \quad (13)$$

Control law:

$$u = u_P + u_I + u_D \quad (14)$$

$$u = K_P e + K_I \int e dt + K_D \frac{de}{dt} \quad (15)$$

Discretization:

$$u(kT) = K_P e(kT) + u_I[(k-1)T] + \frac{K_I T}{2}\left(e[(k-1)T] + e(kT)\right) + K_D \frac{e(kT) - e[(k-1)T]}{T} \quad (16)$$

$$u(kT) = y(1)e(kT) + u_I[(k-1)T] + \frac{y(2)T}{2}\left(e[(k-1)T] + e(kT)\right) + y(3)\frac{e(kT) - e[(k-1)T]}{T} \quad (17)$$

Combining the advantages of two Neural and PID controllers gives us a hybrid Neural-PID controller, the Neural-PID stabilizer has both the stability of a PID controller and the adaptive learning of a Neural controller. The weights are updated immediately after each input and each output are presented. This training method allows the network to learn on-line while the system is operating. Therefore, any new fluctuations that arise in the system are updated. The advantage of this training method is that the network is more adaptive and stable when the system changes over time. The disadvantage is that the network is unstable in the beginning.

3.2.2 Design of RBF-PID controller

The block diagram of the control system is similar to Figure 4(b). In which the feed-forward Neural controller is replaced by the RBF network as follows (Zeng et al., 2012):

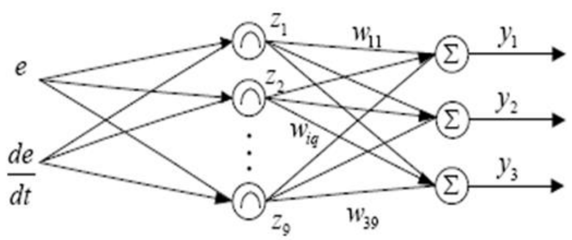


Figure 5. RBF controller with three outputs

The controller uses an RBF neural network with nine basis functions in the hidden layer, the input sum function of the hidden layer neuron is a demand function (Hoori & Motai, 2017). The output has three neurons to adjust three parameters K_P, K_I, K_D. The input sum function of the output layer neuron is a linear function. The input layer consists of two neurons receiving signals from *e* and *de/dt.*

The mathematical equation for transmitting signals from the input layer to the output layer of the network is:

- Output of the *q*th neuron (hidden layer):

$$z_q = e^{\frac{-\|x-\mu_q\|^2}{2\sigma_q}} \quad (18)$$

$$\|x-\mu_q\| = \sqrt{(x-\mu_q)^T(x-\mu_q)} \quad (19)$$

x: input vector consisting of 2 neurons receiving signals from *e* and *de*/*dt*.

μ_q: center of *q*th basis function.

σ_q: width of *q*th basis function.

Output of *i*th neuron (output layer):

$$y_i = \sum_{q=1}^{l} w_{iq} z_q \ (i=\overline{1,n};\ q=\overline{1,l}) \quad (20)$$

Error function:

$$E = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2 \quad (21)$$

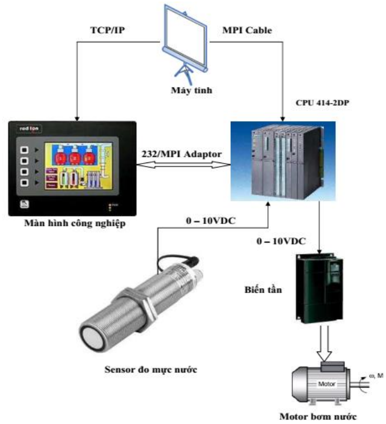
RBF-PID networks are usually trained in two steps: determining the center and width of the basis function. Next is training the output layer weights.

4. RESULTS

4.1 Actual model

The actual control system diagram is shown in Figure 6(a), 6(b). The computer will access the industrial screen via TCP/IP address to view data, graphs and display report data. The 6-inch color industrial screen Touch Screen (Redlion) is used to collect, manage data, change setting values and draw graphs for the control process. The SM431 module is used to receive signals from the water level sensor and compare them with the signal set from the industrial screen. Through CPU 414, the central processor of the PLC S7-400 controller, it is used to write control sub-functions including Neural-PID function, RBF-PID function and output control signals through the SM432 module. The SM432 module is used to output analog signals from 0 to 10VDC to control the inverter to change the speed of the motor pumping water into the tank. The model uses a 3-phase 230VAC water pump, 220W capacity, 50HZ. The 3-phase 230VAC inverter, 0.37KW capacity is used to control the water pump to change speed from 0Hz-50Hz, the output of this signal is decided by the S7-400 controller, the inverter is only the actuator. The software used is STEP 7 V5.5, SCL programming for the controller, programming the screen with Crimson 3.0 software.

(a)



(b)

Figure 6. (a) Actual system (b) Connection diagram

4.2 Measurement results with the hybrid controller NEURAL-PID

Initially set the setpoint to 125 after setting change setpoint to 150, when reaching set change setpoint to 100. The steady-state error e_{xl} is close to 0.5% (this error is caused by surface noise, which is acceptable). Figure 7 is the result of executing the PID neural controller model. At the stage of Setpoint at 125, the result according to the graph in Figure 7 shows a relatively large delay. However, the slope is straight, not overshooting. When changing Setpoint from 125 to 150, the system still reacts faster because at this time the Neural network has been learned. At the stage of changing Setpoint from 150 to 125, there is still an error but it is small and acceptable (the error is caused by model noise). But after that, the system still operates very stably. The ability to respond to the changing process of the system is good. The parameters (connection weights) of the model are trained and updated well. The Neural-PID function controller combines the stability of PID and the adaptive learning of the Neural controller for good control results.

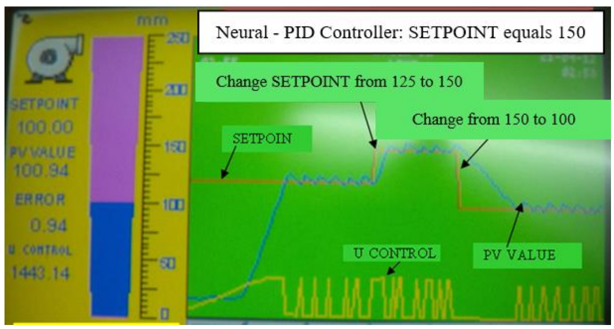


Figure 7. Neural-PID control results with changes in Setpoint up and down

4.3 Measurement results with RBF-PID hybrid controller

Initially set the setpoint to 100, after setting, change the setpoint to 150. The setpoint error e_{xl} is close to 0.5% (this error is caused by surface noise, which is acceptable).

The RBF-PID controller gives the model execution results as shown in Figure 8. We see that the Setpoint stage at 100 has quite good control results but overshoots the first cycle by 20% and then decreases by 10% in cycle 2. It only stabilizes in cycle 3. However, in the steady state mode, the disturbance is less. When changing the Setpoint from 100 to 150, the system still gives an error when changing but is still good afterwards. The error of the model control process is small and acceptable (the error is caused by the model's disturbance). The system's ability to respond to the changing process is quite good. The parameters (connection weights) of the model are well trained and updated. The RBF-PID controller combines the stability of the PID and the adaptive learning of the RBF controller, so it gives good control results. From the measurement results of the two hybrid controllers Neural-PID and RBF-PID in Figure 7 and Figure 8, it shows that both controllers are moving towards stability. If we consider the time to reach stability, the hybrid controller Neural-PID gives better response than the RBF-PID controller.

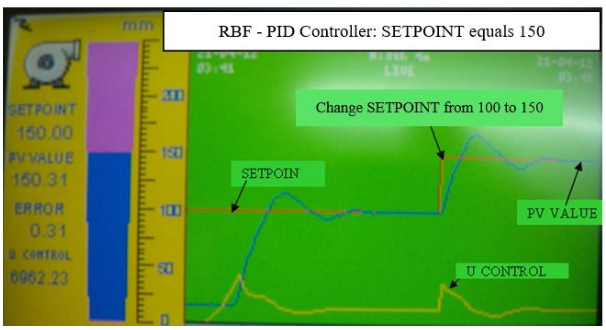


Figure 8. RBF-PID control results with Setpoint changes

4. CONCLUSION

Two types of hybrid controllers Neural-PID and RBP-PID have been performed in the S7-400 system using the SCL programming language, and have shown good performance in controlling the real model of a single water tank. The errors in the test cases are small, with the coefficients *Kp*, *Ki*, *Kd*, center and width of the basis function determined based on the user's experience. The combination of two Neural controllers and a PID controller in the Neural-PID hybrid controller benefits from both technologies: the stability of PID and the learning ability of Neural, leading to better control results in the two tested methods. The advantage of this topic is that the online control using the SCL language in the S7-400 controller has the ability to be used in industry and the application to control industrial systems is very high. But the limitation of the topic is that the language and functions used are only written in Siemens S7-300 and S7-400 controllers. The development direction of the topic is to expand these control functions to other controllers such as Omron, Mitshubisshi, AB, ... and build many other functions with higher practicality and application scope.

REFERENCES

Ahmad, B., & Prajitno, P. (2020). Design of neural network and PLC-based water flow controller. In *Journal of Physics: Conference Series* (Vol. 1528, No. 1, p. 012065).

Al Gizi, A. J., Mustafa, M. W., Al Zaidi, K. M., & Al-Zaidi, M. K. (2015). Integrated PLC-fuzzy PID Simulink implemented AVR system. *International Journal of Electrical Power & Energy Systems*, *69*, 313-326.

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G.,

... & Kindermans, P. J. (2019). iNNvestigate neural networks!. *Journal of machine learning research*, *20*(93), 1-8.

Arora, Y., Singhal, A., & Bansal, A. (2014). A study of applications of RBF network. *International Journal of Computer Applications*, *94*(2).

Berger, H. (2012). *Automating with STEP 7 in STL and SCL: SIMATIC S7-300/400 programmable controllers*. John Wiley & Sons.

Coelho, O., Pires, R., Ferreira, A. S., Gonçalves, B., Alkhoori, S. A., Sayed, M. A., ... & Stocker, J. (2020). The Arabic Version of the personality inventory for the DSM-5 (PID-5) in a clinical sample of United Arab Emirates (UAE) Nationals. *American journal of health behavior*, *44*(6), 794-806.

Combaluzier, S., Gouvernet, B., Menant, F., & Rezrazi, A. (2016). Validation of a French translation of Krueger's personality inventory for DSM-5 in its brief form (PID-5 BF). *L'encephale*, *44*(1), 9-13.

Golenkov, V. V., Korolyov, V. G., Solovyov, A. S., & Tatarenko, V. A. (1992, October). The project of a parallel computer for hardware support of processing semantic and neural networks. In *[Proceedings] 1992 RNNS/IEEE Symposium on Neuroinformatics and Neurocomputers* (pp. 623-634). IEEE.

Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, *8*(2), 1888.

Hoori, A. O., & Motai, Y. (2017). Multicolumn RBF network. *IEEE transactions on neural networks and learning systems*, *29*(4), 766-778.

Ma, D., Song, M., Yu, P., & Li, J. (2020). Research of RBF-PID control in maglev system. *Symmetry*, *12*(11), 1780.

Rossomando, F., & Soria, C. (2015). Design and implementation of adaptive neural PID for non linear dynamics in mobile robots. *IEEE Latin America Transactions*, *13*(4), 913-918.

Topalova, I., & Tzokev, A. (2010). Automated texture classification of marble shades with real-time PLC neural network implementation. In *The 2010 international joint conference on Neural networks (IJCNN)* (pp. 1-8). IEEE.

Webb, A., Davies, S., & Lester, D. (2011). Spiking neural PID controllers. In *Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III 18* (pp. 259-267). Springer Berlin Heidelberg.

Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, *102*, 1645-1656.

Zeng, S., Hu, H., Xu, L., & Li, G. (2012). Nonlinear adaptive PID control for greenhouse environment based on RBF network. *Sensors*, *12*(5), 5328-5348.

Yu, W., & Rosen, J. (2013). Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Transactions on cybernetics*, *43*(2), 673-684.