Embedding the Radial Basis Function Neural Network based PID Controller in a Microcontroller for Controlling the cart's Position in Real-time

Hoang-Dung Nguyen
Faculty of Automation Engineering
Can Tho University
Vietnam

Thanh Dai Le
Faculty of Automation Engineering
Can Tho University
Vietnam

Tuan Kiet Tran
Faculty of Automation Engineering
Can Tho University
Vietnam

The Hien Huynh
Faculty of Automation Engineering
Can Tho University
Vietnam

Proportional integral derivative (PID) is a classic conventional controller and widely utilized in industry. However, its coefficients are normally chosen by using empirical methods. If the parameters of the plant are changed in time or affected by uncertain noises, the conventional controller is not stable because of its fixed coefficients. Therefore, this paper proposes a method to embed the radial basis function neural network based PID controller in the STM32F4VE microcontroller to control the cart's position. Its coefficients are estimated in real-time using the Gradient Descent approach and the radial basis function neural network. Two controllers utilized to control the cart added 2.5 kg payload and 20cm movement. The conventional PID controller made the overshoot of 29.5 % while the proposed method is 2.2 %. The experimental results show that the proposed method can perfectly control the cart's position with the movement distance and the cart's payload changed as well.

Keywords: *adaptive control, embedded system, microcontroller, proportional integral derivative, position control, neural network, radial basis function*

1. INTRODUCTION

Proportional integral derivative (PID) controller is widely utilized in both daily life and industry. This controller is really classis and easy to find the coefficient set based on the experimental method [1, 2]. The coefficients of the PID approach can be obtained by the actual open-loop or closed loop responses [3], trial-error, or estimate these coefficients based on the fuzzy logic approach [4-7], neural networks [1, 8, 9], genetic algorithm [6, 10].

It is a real challenge to determine the PID's coef–ficients to be in accord with the variety of objects [11, 12]. In real world, the PID algorithm with fixed coefficients could not work for uncertain objects such as changing model parameters, model errors, or unknown external disturbance components [1, 13-15]. In this case, the controller's coefficients need to be adjusted online so that the object can be controlled well [7, 9, 15-17].

Neural network is a complicated algorithm with operation functions imitated the human brain. It consists of one input layer, at least one hidden layer, and one output layer. This network has two kinds of the typical process with back-propagation and forward-propagation. The forward neural networks are simpler than back-propagation ones. The connection weights are the most important component for the forward neural networks. They are calculated and changed during training and execution. For back-propagation neural networks, the cost function has to be computed based on the least square error method and Gradient Descent approach to online update weights such that the estimated output is the same as the desired output. The neural network could be combined with other typical approaches such as a model prediction controller for controlling the nonlinear model [1, 18], back-stepping controller for driving cardiac pacemakers [19], and the exact output response prediction [20].

The radial basis function neural network (RBFNN) is an artificial neural network [21]. The radial basis functions are utilized as an activation function. It works as the multi-dimensional function approximation [22]. Also it consists of three layers: one input layer, one hidden layer is designed using the radial basis functions, and one output layer with a linear function. There are many applications are applied with the RBFNN: identification, classification, and estimation [23-26].

The fuzzy-logic based PID method can be utilized to control both linear and nonlinear objects [4-7, 27]. Also, the PID approach with its coefficients estimated using the neural network or particle swarm optimization could be utilized to control nonlinear MIMO (multiple-input multiple-output) objects with many uncertain compo–nents [1, 28-29]. Furthermore, the RBFNNs with rein–forcement learning algorithm are suitable for nonlinear object control as well [30].

Although the radial basis function neural network based PID controller (RBF-PID) could be implemented for real-time applications [19, 31, 32], however, embed–ding it in low-cost controllers has been less interested. Recently, the RBF-PID controller has been designed on the MATLAB software and sends com–mands to an arduino microcontroller board for control–ling the carriage [8]. Also, the PID controller with embedded

Received: November 2024, Accepted: December 2024
Correspondence to: Dr Hoang-Dung Nguyen, Faculty of Automation Engineering, Can Tho University; Campus II, 3/2 street, Ninh Kieu district, Can Tho city, Vietnam
E-mail: hoandung@ctu.edu.vn
doi: 10.5937/fme2501063N

self-tuning algorithm in FPGA (Field prog–rammable gate array) IC (integrated circuit) and algorithms are developed using the VHDL (very high speed integrated circuit hardware description language) language [13].

Therefore, the objective of the current work is to embed the RBFNN based PID algorithm in a microcontroller to control the nonlinear object. The PID's coefficients are estimated by the RBFNN with the Gradient Descent approach. To demonstrate the proposed approach implemented in the real word, both the conventional PID and proposed RBF-PID approaches are embedded in a STM32F4VE chip for controlling the cart's position in real-time. The experimental results of the proposed RBF-PID can compare with the conventional PID approach. Additionally, to avoid the fluctuation of the cart during its movement, the S-curve acceleration control approach is applied.

2. METHOD

2.1 PID controller

The typical PID controller comprises the proportional, integral, and derivative components shown as follows [11, 33].

$$G_{PID}\left(s\right)=K_p\left(1+\frac{1}{T_i s}+T_d s\right) \tag{1}$$

where, K_p is a proportional component. $T_i=\frac{K_p}{K_i}$ and $T_d=\frac{K_p}{K_d}$ refer constant time values determined by proportional (K_p), integral (K_i) and derivative (K_d) coefficients. Therefore, this approach is built by addressing the set of coefficients {K_p, K_i, K_d} of the equation (1) to ensure conditions on quality indicators.

2.2 PID controller design based on Zeigler-Nichols method

The empirical procedure utilized to build the PID controller is developed by [3]. There are two ways to address the coefficient set of the control law from the open- and closed-loop system responses. For the open-loop system response (see Figure 1), they are calculated with the L delay time and T constant time shown in Table 1. The second way is to employ the cycle T_c of the feedback system response to figure out the controller's coefficients (see Table 2).

Table 1. Addressing the coefficients of the PID controller based on the *L* delay time and the *T* time constant.

Controllers	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	$0.5L$

Table 2. Addressing the coefficients of the PID controller based on the set-point *K* and the T_c period.

Controllers	K_p	T_i	T_d
P	$0.5K$	∞	0
PI	$0.45K$	$0.85T_c$	0
PID	$0.6K$	$0.5T_c$	$0.125T_c$

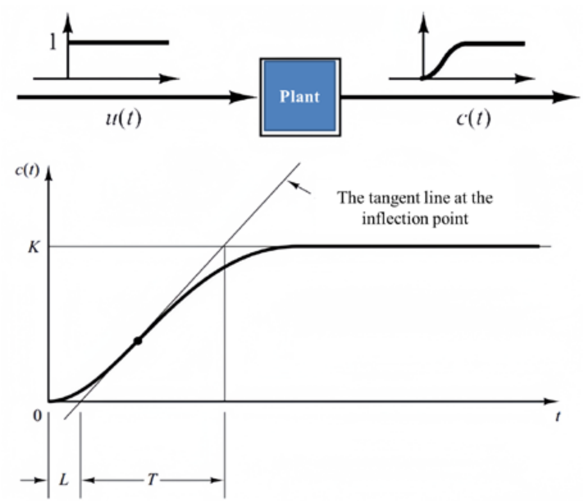


Figure 1. The open loop response (*c*(*t*)) respected to the step function input signal (*u*(*t*)): *L* refers the delay time, *T* stands for the time constant, and *K* is the set-point value.

2.3 Radial basis function neural network

The RBF neural network (RBFNN) has been widely applied because of their simplicity and great learning speed. It is based on approximating a multidimensional function [16]. Its structure comprises input, hidden, and output layers (see Figure 2); RBFNN's input is described as $X=\left[x_1 \quad x_2 \quad \cdots \quad x_n\right]^T$; T is the transposition; n refers the neuron quantity of the hidden layer. $H=\left[h_1 \quad h_2 \quad \cdots \quad h_m\right]^T$ signifies the basis vector for this layer; $h_j=\exp\left(-\frac{\left\|X-C_j\right\|^2}{2b_j^2}\right), j=1,2,...,m$ indicates the j^{th} Gaussian function [19]; m is the neuron quantity of the hidden layer; $C_j=\left[c_{j1} \quad c_{j2} \quad \cdots \quad c_{jn}\right]^T$ denotes the basis center at the j^{th} node; $B=\left[b_1 \quad b_2 \quad \cdots \quad b_m\right]^T$ signifies the basis width, $b_j > 0$ is the width of the basis function at the j^{th} node; $W=\left[w_1 \quad w_2 \quad \cdots \quad w_m\right]^T$ describes the output weight vector. The total weight of the RBFNN is calculated as follows.

$$y_m(k)=W^T h=w_1h_1+w_2h_2+\cdots+w_mh_m \tag{2}$$

where, $y_m(k)$ means the estimated response and calculated at the k^{th} time step. The weights of the RBFNN are updated by the Gradient Descent algorithm consisting of weight ($\Delta w_j(k)$), center (($\Delta c_{ji}(k)$), and width (($\Delta b_j(k)$) as follows.

$$\Delta w_j(k)=\eta\left(y(k)-y_m(k)\right)h_j \tag{3}$$

where, $y(k)$ refers the actual response of the controlled object.

$$\Delta b_j(k)=\eta\left(y(k)-y_m(k)\right)w_jh_j\frac{\left\|X-C_j\right\|^2}{b_j^3} \tag{4}$$

$$\Delta c_{ji}(k)=\eta\left(y(k)-y_m(k)\right)w_j\frac{x_j-c_{ji}}{b_j^2} \tag{5}$$

where, η is the momentum coefficient; $w_j(k)$, $b_j(k)$, and $c_{ji}(k)$ are updated as follows.

$$w_j(k) = w_j(k-1) - \Delta w_j(k) + \alpha\left(w_j(k-1) - w_j(\text{k}-2)\right) \quad (6)$$

$$b_j(k) = b_j(k-1) - \Delta b_j(k) + \alpha\left(b_j(k-1) - b_j(k-2)\right) \quad (7)$$

$$c_{ji}(k) = c_{ji}(k-1) - \Delta c_{ji}(k) + \alpha\left(c_{ji}(k-1) - c_{ji}(k-2)\right) \quad (8)$$

where, α stands for the learning rate and Jacobian is computed as follows.

$$\begin{aligned}\frac{\partial y(k)}{\partial \Delta u(k)} &\approx \frac{\partial y_m(k)}{\partial \Delta u(k)} = \frac{\partial \sum_{j=1}^{m} w_j h_j}{\partial \Delta u(k)} \\ &= \frac{\partial \sum_{j=1}^{m} w_j \exp\left(-\frac{\left\|X - C_j\right\|^2}{2b_j^2}\right)}{\partial \Delta u(k)} \\ &= \frac{\sum_{j=1}^{m} w_j h_j \frac{C_j - X}{b_j^2} \partial X}{\partial \Delta u(k)} = \sum_{j=1}^{m} w_j h_j \frac{C_j - X}{b_j^2}\end{aligned} \quad (9)$$

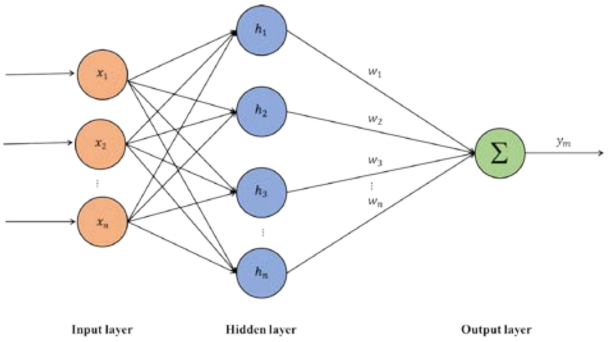


Figure 2. The structure of the RBFNN.

2.4 RBFNN based PID controller

Figure 3 shows the PID algorithm with coefficients $\{K_p, K_i, K_d\}$ estimated using the Jacobian function of the RBFNN. The difference of the the desired response $y_d(k)$ and the actual response $y(k)$ is determined as follows.

$$e(k) = y_d(k) - y(k) \quad (10)$$

$$x_c(1,k) = e(k) - e(k-1) \quad (11)$$

$$x_c(2,k) = e(k) \quad (12)$$

$$x_c(3,k) = e(k) - 2e(k-1) + e(k-2) \quad (13)$$

The control law of the RBF-PID, $\Delta u(k)$ in which its coefficients, $\{K_p, K_i, K_d\}$ are estimated in real-time using the RBFNN as follows.

$$\begin{aligned}\Delta u(k) = K_p(k)\left[e(k) - e(k-1)\right] + K_i(k)e(k) + \\ K_d\left[e(k) - 2e(k-1) + e(k-2)\right]\end{aligned} \quad (14)$$

The coefficients of the RBF-PID controller are updated based on the least squared error method $E(\text{k}) = \frac{1}{2}e(k)$ and the Gradient Descent algorithm.

$$\begin{aligned}\Delta K_p(k) = -\eta\frac{\partial E}{\partial K_p} = -\eta\frac{\partial E}{\partial y}\frac{\partial u}{\partial \Delta u}\frac{\partial \Delta u}{\partial K_p} = \\ \eta e(k)\frac{\partial y}{\partial \Delta u}x_c(1,k)\end{aligned} \quad (15)$$

$$\begin{aligned}\Delta K_i(k) = -\eta\frac{\partial E}{\partial K_i} = -\eta\frac{\partial E}{\partial y}\frac{\partial u}{\partial \Delta u}\frac{\partial \Delta u}{\partial K_i} = \\ \eta e(k)\frac{\partial y}{\partial \Delta u}x_c(2,k)\end{aligned} \quad (16)$$

$$\begin{aligned}\Delta K_d(k) = -\eta\frac{\partial E}{\partial K_d} = -\eta\frac{\partial E}{\partial y}\frac{\partial u}{\partial \Delta u}\frac{\partial \Delta u}{\partial K_d} = \\ \eta e(k)\frac{\partial y}{\partial \Delta u}x_c(3,k)\end{aligned} \quad (17)$$

where, $\frac{\partial y}{\partial \Delta u}$ is the Jacobian computed from the equ–ation (9).

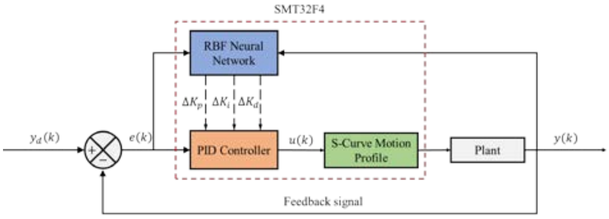


Figure 3. The scheme of the RBF-PID controller.

2.5 Embedding the RBF-PID approach in the STM32F4VE microcontroller

STM32F4VE is a 32 bit ARM cortex MCU micro–controller series (manufactured by STMicroelectronics) designed for medical, industrial, and civil related applications. This microcontroller is designed to be robust with enhanced performance, flash memory up to 1 MHz, SRAM 192KB memory and flexible I/O communicator (2 USB ports, 15 conventional interfaces including USART, SPI, I2C, CAN, and SDIO), up to 17 time counters for both 16 bits and 32 bits. It can calculate the floating point with 32 bits and the oscillation frequency of 168 MHz.

Microcontroller STM32F4VE is programmed using the C language. Therefore, the RBF based PID control algorithm can be embedded in this microcontroller to improve control quality, low design cost, and the ability to apply control in real-time. The RBF-PID controller is built from equations (9)-(17) shown in Figure 3. The red dash box of the Figure 3 shows the diagram of the RBF-PID controller integrated in the STM32F4VE. The control law is built based on the equation (14). The coefficients $\{K_p, K_i, K_d\}$ of the RBF-PID controller are updated in real-time shown in the equations (15)-(17). In order to avoid the case of the response being overshoot at the beginning and end duration, this study developed an acceleration control approach using the S-curve motion profile (see Section 2.6).

The RBF-PID control algorithm developed from equations (9)-(17) is programmed using the C language with the Keil C software. This algorithm is compiled into binary code and embedded in the STM32F4VE microcontroller. When starting, the cart will be moved back to the home position. The desired position in which the cart needs to move is set from the computer and sends to the STM32F4VE microcontroller. At this time, the microcontroller reads the current position of the cart through the encoder and estimates the coef–ficients $\{K_p, K_i, K_d\}$ of the RBF-PID controller using the Gradient Descent approach. The RBFNN utilized in this work consists of 3 layers (see Figure 2): (1) the input layer is the cart's actual position $y(k)$ and the position error signal $e(k)$; (2) the hidden layer with 5 nodes is computed by $h_j = \exp\left(-\frac{\|X - C_j\|^2}{2b_j^2}\right), j = \overline{1,5}$; (3) the output layer is a linear equation updated by equations (2) and (3) using the Gradient Descent approach. The control law is generated under the pulse width modu–lation (PWM) method combined with the acceleration control approach (see the following Section 2.6) to ensure that the cart moves smoothly and reaches the desired position.

2.6 Acceleration control approach

Acceleration control approach based on the S-curve motion profile is a speed control technique that ensures the process of changing speed is smooth and stable. Figure 4 depicts the position control (upper panel) and speed control (lower panel). The goal of the acceleration control is to avoid sudden changes in speed that cause the cart to move incorrectly to the desired position. The S-curve-based acceleration control method consists of three stages: steadily increasing acceleration, uniform acceleration, and steadily decreasing acceleration.

Suppose that, the total number of pulses of the movement to reach the desired position is N. The acceleration control process is divided into three stages and based on the total number of pulses to drive the cart to the desired position.

The early stage ($0 \le t < n_1$) is a steadily increasing acceleration movement with the pulse n_1, t is the time step. At this stage, the acceleration $a(t)$ changes linearly according to the following expression.

$$a(t) = \mathrm{a}_0 + \mathrm{q}t \tag{18}$$

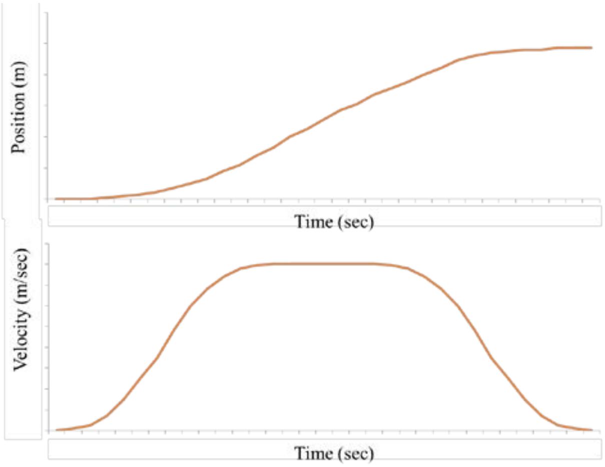


Figure 4. The S-curve profile applied for controlling the cart's acceleration.

where a_0 is the initial acceleration, q is the acceleration constant. And the velocity is calculated as follows.

$$v(t) = \mathrm{v}_0 + \mathrm{a}_0 t + \frac{\mathrm{q}t^2}{2} \tag{19}$$

where, v_0 is the initial velocity.

The second stage ($n_1 \le t < n_2$) is the period of acceleration stability in the period from n_1 to n_2 pulse. At this point $t = n_1$, the acceleration reaches its maxi–mum value and keeps the acceleration constant $a(n_1) = a_{\max} = \mathrm{q}n_1 + a_0$ and velocity is $v(n_1) = (\mathrm{q}n_1 + a_0)n_1 + (k(n_2 - n_1)^2)/2 + v_0$.

The third stage ($n_2 \le t < n_3$) has a steady decreasing acceleration when it reaches the $t = n_2$. At this time, acceleration and velocity are $a(n_2) = a_{\max} – \mathrm{q}n_2$ and $v(n_2) = v_0 + a_{\max}n_2 + (\mathrm{q}(n_3 - n_2)^2)/2$, respectively.

2.7 Controlling the cart's position

Position control model

Figure 5(a) shows components of the position control model. The key components of this model consist of a cart (left and right movement) and a gear belt connected a NF5475 DC motor with built-in encoder (voltage supply 12-24 VDC, power 32 W, maximum speed 4,500 rpm, manufacturer Nisca, Japan) to move the cart. The cart is permanently connected to the gear belt. A relative encoder attached on the motor allows its speed/position based on the number of pulses to be read. The resolution of the encoder is 200 pulses/ring. These pulses are sent back to the STM32F4VE microcontroller to calculate the desired speed, acceleration, and movement distance. The microcontroller gives the commands to control the NF5475 motor through the BTS7950 driver circuit (1,000W maximum power). Table 3 presents the para–meters of the position control model. Figure 5(b) illus–trates the facilities utilized for implementing the expe–riment on controlling the cart's position in real time.

The encoder output is connected to two analog inputs (A0 and A1) of the STM32F4VE microcontroller for reading the pulses using the T2 timer. Depending on the number of pulses, the position/distance of the cart is accurately determined by the RBF-PID algorithm. The output of this controller consists of a rotation (controlled via pins C8 and C9 of the STM32F4VE micro–controller) and a pulse width modulation (PWM) method connected to the input of the S-curve profile acceleration control algorithm (see Section 2.6).

Table 3. Specification of the position control model.

Description	Value
Model size (Length × Width × Height) [cm]	70×30×20
Motion distance [cm]	0 - 35
Cart (Length × Width × Height) [cm]	15 × 4 × 5
Rear belt (Length × Width × Height) [cm]	58 × 1 × 0.1
Weight [kg]	12.5

The output of the S-curve profile algorithm is utilized to control the NF5475 motor through the BTS7960 driver. The desired location and display the actual location of the cart in real-time are set from the computer through the SMT32F4VE's SWD (Single-wire debug) protocol. Figure 6 shows the hardware circuit diagram connecting the STM32F4VE microcontroller to the peripherals: limit switches/sensors, reading the pulses from encoder for controlling the cart's position, BTS7950 driver for controlling the NF5475 motor. Figure 7 depicts scheme for controlling the cart position in real time.

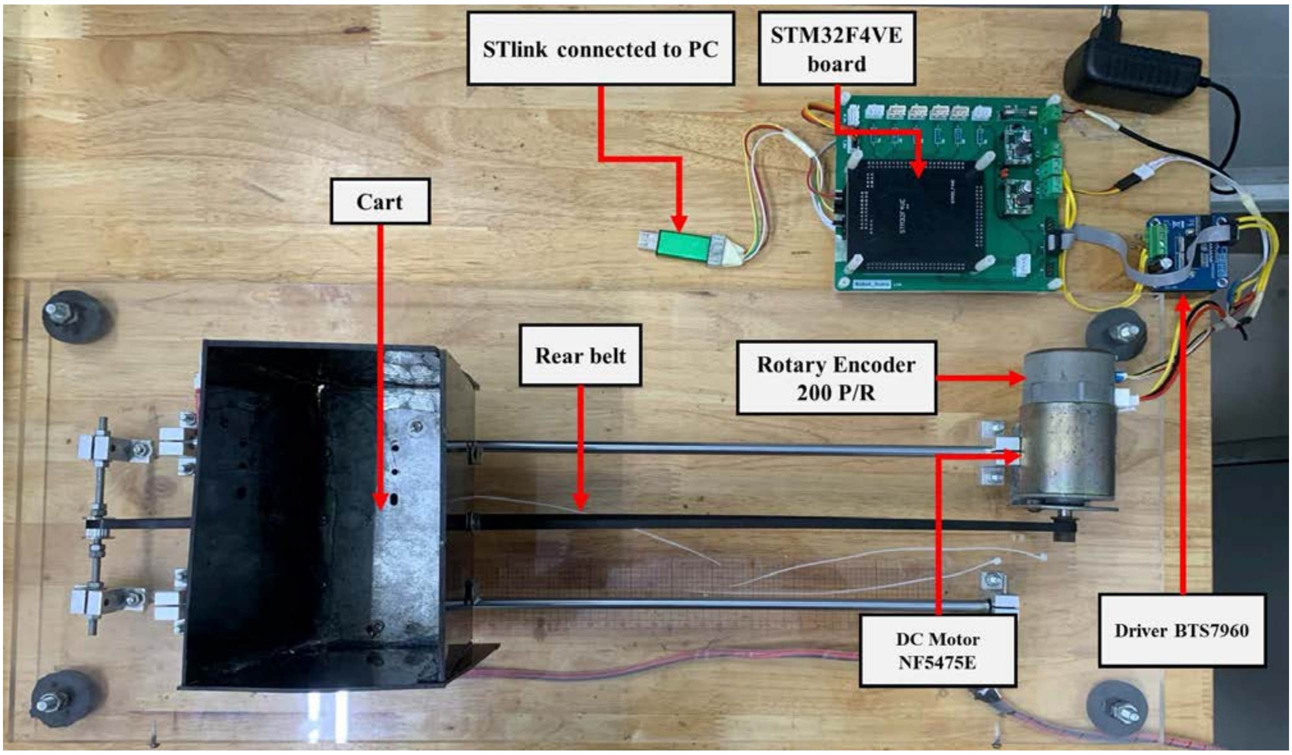
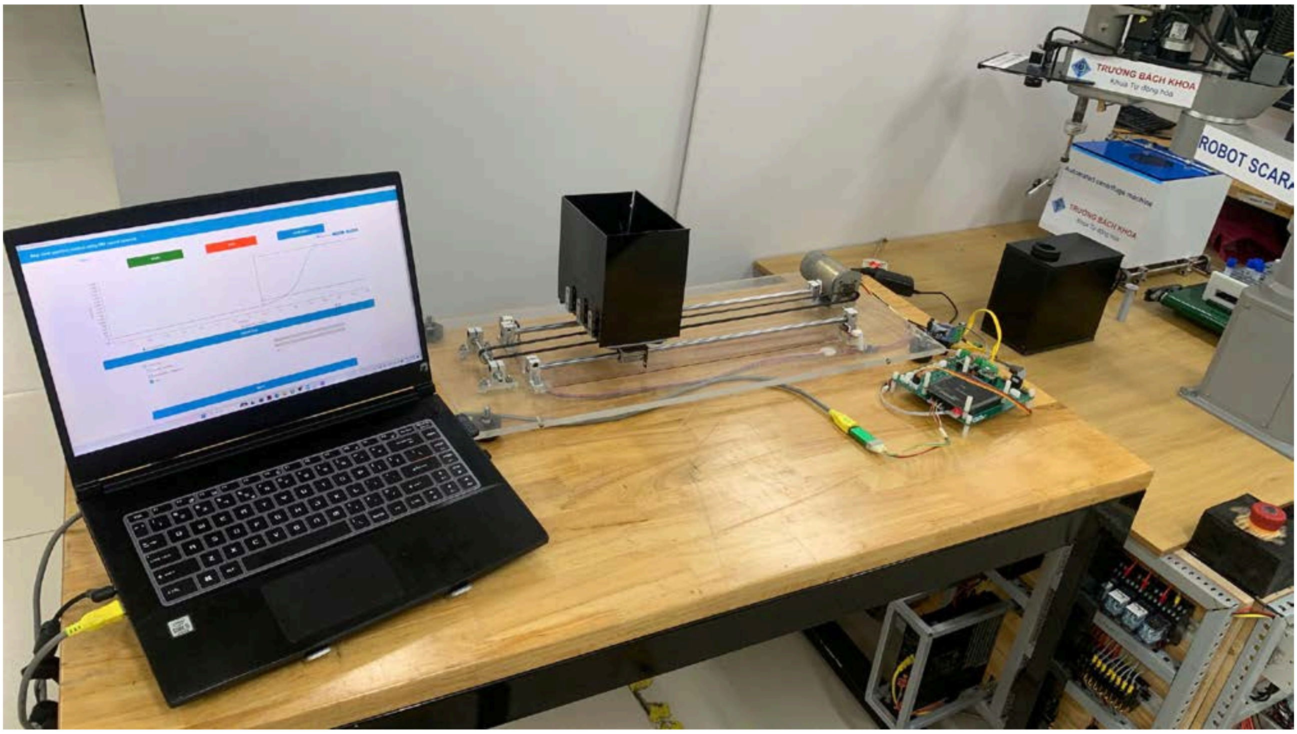


Figure 5 (a) Components of the position control model.



(b) The cart controlled by the RBF-PID embedded in the STM32F4VE microcontroller.

Figure 5. Position control model: (a) Components of the position control model and (b) The cart controlled by the RBF-PID embedded in the STM32F4VE microcontroller (the cart's response, the RBF-PID control law, and estimated coefficients are monitored on the computer screen)

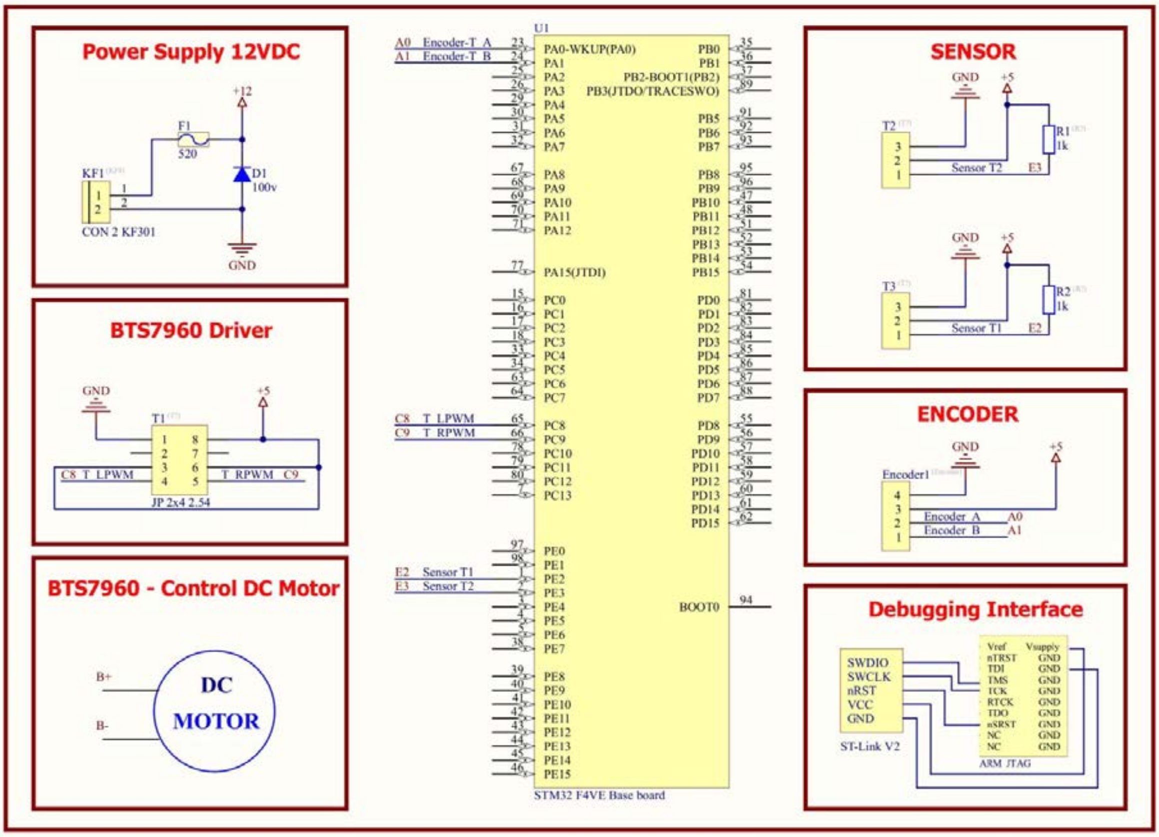


Figure 6. Hardware schematic of STM32F4VE for controlling the cart.

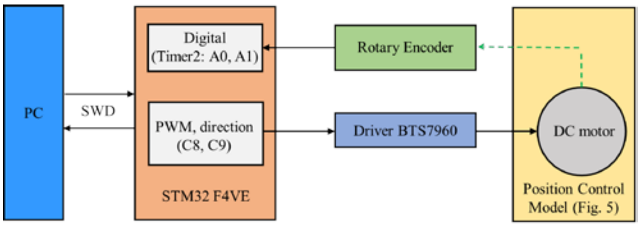


Figure 7. Scheme for controlling the cart position in real time.

The relationship between the distance and the number of control pulses

The encoder used in this study has the resolution of the 200 pulses/ring. It is attached directly on the shaft of the NF5475 DC motor with a circumference $2\pi R$ (R is the radius of the pulley attached directly to the shaft of the motor). For each pulse, the motor can move at an angle as follows.

$$1\text{pulse} = \frac{360^0}{200} = 1.8^0 \qquad (20)$$

Therefore, the cart's travel distance, S is calculated as follows.

$$S = \frac{p \times 1.8^0}{360^0} \times 2\pi R \qquad (21)$$

where p is the number of pulses to be controlled.

3. RESULTS AND DISCUSSION

In order to evaluate the effectiveness of the RBF-PID controller in real-time, two experiments were carried out and compared with the conventional PID controller: (1) cart's position control in the case of the cart without payload; (2) cart's position control in case of carrying payload (1.0 kg and 2.5 kg).

For the conventional PID controllers, the Zeigler-Nichols method was applied to determine the coefficients $\{K_p, K_i, K_d\}$ (see Section 2.2) and the controller is designed using equation (1). It is worthy note that the cart's position control algorithm using the conventional PID controller is also embedded in the STM32F4VE microcontroller. In this study, the coefficients of the PID controller were selected as follows $\{K_p = 3.8, K_i=1.5, K_d = 0.25\}$.

For the RBF-PID control algorithms, the W-weights are randomly initialized and updated after each iteration, the basis function width b = 40 cm, its center c = 10 cm, the momentum coefficient $\eta = 0.2$, and the learning rate $\alpha = 0.15$. Notably, the RBF-PID controller's coefficients are continuously updated in real-time using the equations (14)-(17) to ensure that the cart's position controlled to reach the desired position.

In order to ensure the cart with the smooth and inertial movement, the acceleration control algorithm is applied for both the PID and RBF-PID controllers (see Section 2.6). In the first stage, the acceleration gradually increases to 1/4 of the total pulses from the starting position to the desired position. The movement distance is calculated using the equation (21). The second stage is the acceleration stabilization phase when the cart moves from 1/4 to 3/4 of the total pulses. The final stage is a decreasing steady acceleration from 3/4 of the total pulses to the desired position. a_0 = 0.001 mm/s^2 is the initial acceleration and v_0 = 0.2 mm/s refers the initial velocity calculated for the equations (18) and (19).

Controlling the cart’s position without payload

Both the conventional PID controller and the proposed RBF-PID controller are utilized to control the position of the cart without payload. Figure 8 (in the upper panel) shows the position of the cart controlled by the conventional PID controller (blue dash curve) and the proposed RBF-PID controller (red solid curve) relative to the desired position y_d (dotted black line). The experimental results have proven that the position output of the cart controlled by the two controllers tracking to the desired position with a zero approximate error, no overshoot and a quick settling time of 0.8 s. Figure 8 (in the lower panel) depicts the control laws of the conventional PID controller (magenta dash curve) and the proposed RBF-PID controller (green solid curve). It can be seen that at the time of sudden position changes, the control law also varies accordingly to ensure that the actual movement position meets the desired position. Both controllers are quite similar. Coefficients $\{K_p, K_i, K_d\}$ of the proposed RBF-PID controller are continuously updated in real-time and are shown in Figure 9. Similar to the conventional PID controller, at times of changing the cart’s position, the coefficients of the proposed RBF-PID vary greatly. This helps the controller generates the right decision for driving the cart to the desired position [15].

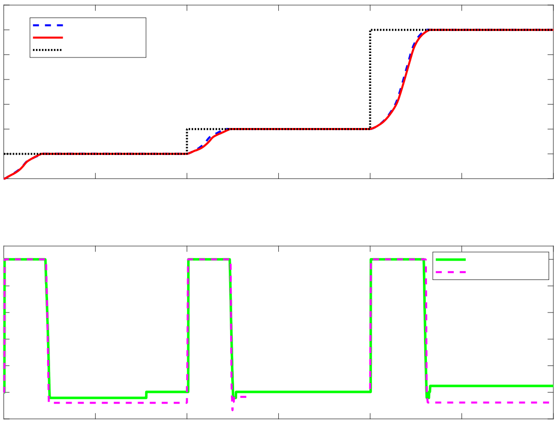


Figure 8. The cart’s position without payload controlled by the conventional PID (blue dash curve) and the proposed RBF-PID (red solid curve) compared with the desired position y_d (dotted black line) in the upper panel and the control law (green solid curve) in the lower panel developed from the equation (14).

Controlling the cart’s position with payload of 1 kg and 2.5 kg

To demonstrate whether the conventional PID controller and the proposed RBF-PID controller being adaptable with the changes of the cart’s parameters, the 1.0 kg and 2.5 kg payloads are added into the cart separately. Figure 10 (in the upper panel) shows the cart’s position with a 1.0 kg payload driven by the conventional PID controller (blue dash curve) and the proposed RBF-PID controller (red solid curve). The experimental results show that, when changing from 500 to 1,000 pulses, the response of the conventional PID has the slower response than the proposed RBF-PID approach of 1.903 s and 1.617 s, respectively. If the cart is moved with the distance further (2,000 pulses), i.e. from 1,000 pulses to 3,000 pulses, the response of the conventional PID controller with the settling time and the overshoot is 5.4 s and 26.7 %, respectively, while the proposed RBF-PID controller with a short settling time of 1.47 s and a negligible overshoot.

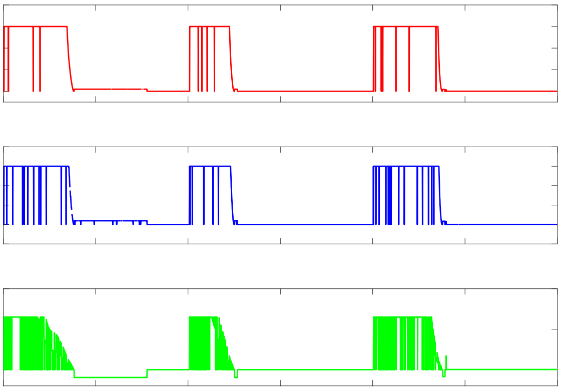


Figure 9. The coefficients of the RBF-PID controller without payload: Δ*Kp* (red solid curve in the upper panel), Δ*Ki* (blue solid curve in the middle panel), and Δ*Kd* (green solid curve in the lower panel) developed from the equations (15)-(17).

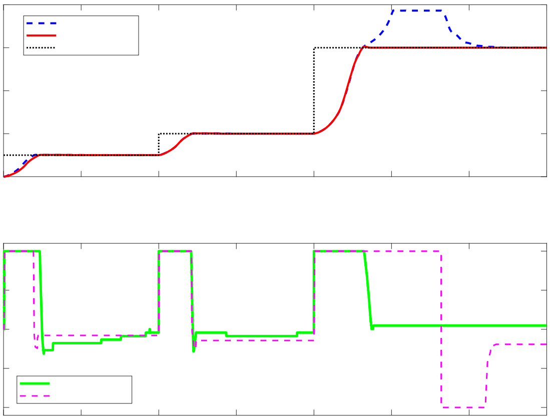


Figure 10. The cart’s position with the 1kg payload controlled by the conventional PID (blue dash curve) and the proposed RBF-PID (red solid curve) compared with the desired position y_d (dotted black line) in the upper panel and the control law (green solid curve) in the lower panel developed from the equation (14).

The control laws of the conventional PID controller (magenta dash curve) and the proposed RBF-PID controller (green solid curve) are presented in the lower panel of Figure 10. It is worthy note that, from the initial time to 9 s, the two controllers still working well. However, at the time of changing the cart’s position from 1,000 to 3,000 pulses (at the time of greater than 9s), the conventional PID controller is largely fluctuated while the proposed RBF-PID controller is not. When the parameters of the object change (in this case, the payload of the cart increases to 1 kg and the movement distance is 20 cm or 2,000 pulses), then the conventional PID approach is out of control while the proposed RBF-PID controller can adapt with those changes according to the adjustment of its coefficients in real-time. The coefficients of the proposed RBF-PID cont–

roller (see Figure 11) are continuously adjusted over time to ensure that the current cart position tracking to the desired position. If the cart's payload is changed to 2.5 kg and the cart moves with a distance of 2,000 pulses, the cart's response using the conventional PID controller has the settling time of 5.3 s and the overshoot of 29.2 % while the proposed RBF-PID controller has a settling time of 1.74 s and the overshoot of 2.2 % (see Figure 12 in the upper panel). Figure 12 (in the lower panel) shows the conventional PID controller (magenta dash curve) and the proposed RBF-PID controller (green solid curve). Figure 13 depicts the coefficient set *Kp*, *Ki* and *Kd* of the proposed RBF-PID controller constantly updated using the RBFNN approach.

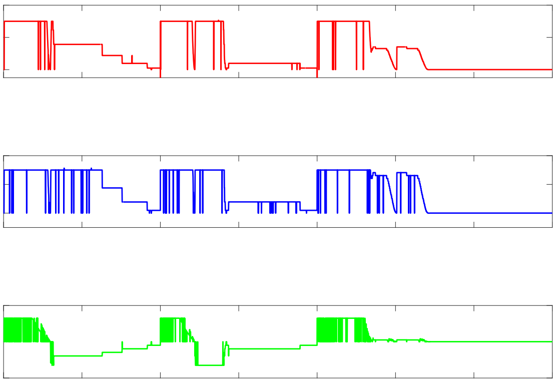


Figure 11. The coefficients of the RBF-PID with the 1 kg payload: Δ*Kp* (red solid curve in the upper panel), Δ*Ki* (blue solid curve in the middle panel), and Δ*Kd* (green solid curve in the lower panel) developed from the equations (15)-(17).

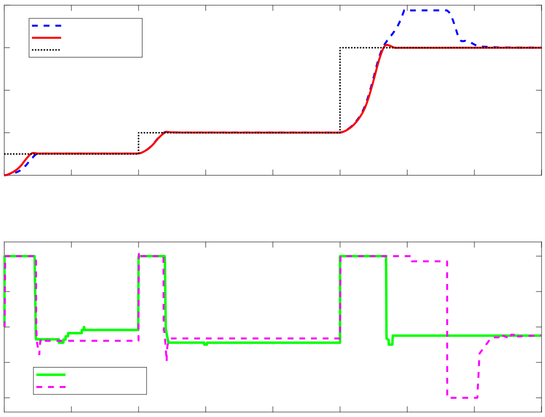


Figure 12. The cart's position with the 2.5 kg payload controlled by the conventional PID (blue dash curve) and the proposed RBF-PID (red solid curve) compared with the desired position y_d (dotted black line) in the upper panel and the control law (green solid curve) in the lower panel developed from the equation (14).

In short, if the cart carries a payload and moves a long distance (20 cm), the conventional PID controller is unstable (the overshoot is approximately 30 %) while the proposed RBF-PID controller still ensures stability and the overshoot is negligible thanks to the adjustment of the coefficients in real-time to adapt to the change of the cart's parameters [15]. The proposed RBF-PID controller with the adjustable coefficients is better the conventional PID approach [34]. The obtained results of our work is entirely agreed with the previous works [4, 9, 15-17, 28].

Recently, researchers have been interested in applying the PID controller with estimated coefficients in real-time for driving strong nonlinear systems. The approaches such as balancing composite motion optimization [12], improved particle swarm optimization [14, 29, 35], and RBFNNs [9, 15] have been utilized to ascertain the optimal values of the PID controller. They demonstrated their performance (e.g., adaptive capacity, robustness, stability, and disturbance reduction) for controlling the highly nonlinear objects. Actually, those algorithms are quite complex. Therefore, the simulation is a firstly suitable solution to prove their effectiveness. A few recent works run the algorithms on the computer to send the commands to a microcontroller and get feedback signals for updating the parameters [8, 12]. The microcontroller receives the control signal to drive the nonlinear systems. Also it gets the actual responses from sensors and sends them back to the algorithm on the computer for comparison with the desired signal. In our current work, the RBF-PID approach is entirely embedded in the STM32F4VE microcontroller to control the cart's position in real time. This work is promised for embedding more complicated algorithms in the low cost microcontroller.

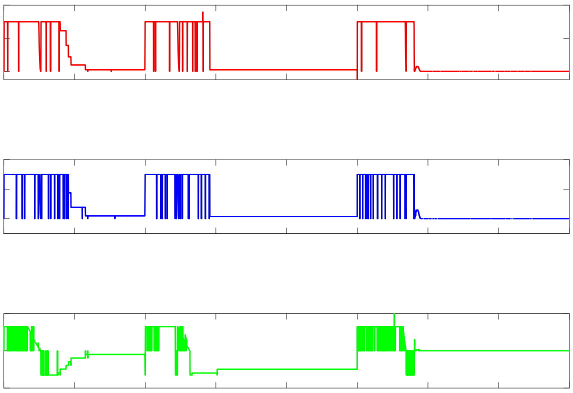


Figure 13. The coefficients of the RBF-PID with the 2.5 kg payload: Δ*Kp* (red solid curve in the upper panel), ΔKi (blue solid curve in the middle panel), and Δ*Kd* (green solid curve in the lower panel) developed from the equations (15)-(17).

4. CONCLUSIONS

This work presented the RBFNN based PID controller embedded in the STM32F4VE microcontroller to drive the position of the cart in the real-time. The parameters $\{K_p, K_i, K_d\}$ of the PID controller were estimated using the RBFNN with Gradient Descent approach. To smoothly and precisely ensure the cart's movement, the acceleration control method with the S-curve profile was applied. The experimental results showed that in the case of the cart without the payload, the performance of both the conventional PID controller and the proposed RBF-PID controller was the same. However, if the cart was added the 2.5 kg payload, the proposed RBF-PID controller had the better performance for both the settling time (1.74 s) and the overshoot (2.2 %) compared with the conventional PID controller, 5.3 s and 29.5 %, respectively. The RBF-PID controller was stable while the conventional one was not. Therefore, our proposed approach promises for developing the smart controller with the low cost and can control the actuator in real-time.

ACKNOWLEDGMENT

We would like to thank Can Tho University for supporting this work.

REFERENCES

[1] S. Cong, Y. Liang, "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems," *IEEE Transactions on Industrial Electronics,* vol. 56, no. 10, pp. 3872-3879, 2009.

[2] M. J. Orr, "Introduction to radial basis function networks," ed: Technical Report, center for cognitive science, University of Edinburgh …, 1996.

[3] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the American society of mechanical engineers,* vol. 64, no. 8, pp. 759-765, 1942.

[4] G. K. Mann, B.-G. Hu, and R. G. Gosine, "Analysis of direct action fuzzy PID controller structures," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 29, no. 3, pp. 371-388, 1999.

[5] W. Z. Qiao, M. Mizumoto, "PID type fuzzy controller and parameters adaptive method," *Fuzzy sets and systems,* vol. 78, no. 1, pp. 23-35, 1996.

[6] K.-S. Tang, K. F. Man, G. Chen, and S. Kwong, "An optimal fuzzy PID controller," *IEEE transactions on industrial electronics,* vol. 48, no. 4, pp. 757-765, 2001.

[7] R. Jovanović, U. Bugarić, M. Vesović, and N. Perišić, "Fuzzy Controller Optimized by the African Vultures Algorithm for Trajectory Tracking of a Two–Link Gripping Mechanism," *FME Transactions,* vol. 50, no. 3, pp. 491-501, 2022.

[8] H.-D. Nguyen, T. H. Huynh, "Controlling the Position of the Carriage in Real-Time Using the RBF Neural Network Based PID Controller," in *2018 18th International Conference on Control, Automation and Systems (ICCAS),* 2018: IEEE, pp. 1418-1423.

[9] X. Shi, H. Zhao, Z. Fan, "Parameter optimization of nonlinear PID controller using RBF neural network for continuous stirred tank reactor," *Measurement and Control,* vol. 56, no. 9-10, pp. 1835-1843, 2023.

[10] C. Harpham, C. W. Dawson, M. R. Brown, "A review of genetic algorithms applied to training radial basis function networks," *Neural Computing & Applications,* vol. 13, pp. 193-201, 2004.

[11] K. H. Ang, G. Chong, Y. Li, "PID control system analysis, design, and technology," *IEEE transactions on control systems technology,* vol. 13, no. 4, pp. 559-576, 2005.

[12] V.-T. Nguyen *et al.*, "Robust adaptive nonlinear PID controller using radial basis function neural network for ballbots with external force," *Engineering Science and Technology, an International Journal,* vol. 61, p. 101914, 2025.

[13] Y.-S. Kung, H. Than, and T.-Y. Chuang, "FPGA-realization of a self-tuning PID controller for X–Y table with RBF neural network identification," *Microsystem Technologies,* vol. 24, pp. 243-253, 2018.

[14] M. Ahmadnia, A. Hajipour, H. Tavakoli, "Robust variable-order fractional PID-LP fuzzy controller for Automatic Voltage Regulator systems," *Applied Soft Computing,* vol. 167, p. 112268, 2024.

[15] N. H. Sahrir, M. A. Mohd Basri, "Radial Basis Function Network Based Self-Adaptive PID Controller for Quadcopter: Through Diverse Conditions," *International Journal of Robotics & Control Systems,* vol. 4, no. 1, 2024.

[16] O. Rodríguez-Abreo, J. Rodríguez-Reséndiz, C. Fuentes-Silva, R. Hernández-Alvarado, M. D. C. P. T. Falcón, "Self-tuning neural network PID with dynamic response control," *IEEE Access,* vol. 9, pp. 65206-65215, 2021.

[17] A. Rospawan, C.-C. Tsai, C.-C. Hung, "Output Recurrent Fuzzy Broad Learning Systems for Adaptive MIMO PID Control: Theory, Simulations and Application," *IEEE Access,* 2024.

[18] N. Perišić, R. Jovanović, "Control of direct current motor by using artificial neural networks in Internal model control scheme," *FME Transactions,* vol. 51, no. 1, pp. 109-116, 2023.

[19] M. E. Karar, "Robust RBF neural network–based backstepping controller for implantable cardiac pacemakers," *International Journal of Adaptive Control and Signal Processing,* vol. 32, no. 7, pp. 1040-1051, 2018.

[20] S. Rehman *et al.*, "Wind Speed Prediction Based on LongShort Term Memory using Nonlinear Autoregressive Neural Networks," *FME Transactions,* vol. 50, no. 2, 2022.

[21] A. G. Bors, I. Pitas, "Median radial basis function neural network," *IEEE transactions on Neural Networks,* vol. 7, no. 6, pp. 1351-1364, 1996.

[22] T. Poggio, F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE,* vol. 78, no. 9, pp. 1481-1497, 1990.

[23] M. J. Er, S. Wu, J. Lu, H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE transactions on neural networks,* vol. 13, no. 3, pp. 697-710, 2002.

[24] K. Narendra, V. Sood, K. Khorasani, R. Patel, "Application of a radial basis function (RBF) neural network for fault diagnosis in a HVDC system," *IEEE transactions on power systems,* vol. 13, no. 1, pp. 177-183, 1998.

[25] J. Fei, T. Wang, "Adaptive fuzzy-neural-network based on RBFNN control for active power filter," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 1139-1150, 2019.

[26] P. Dhanalakshmi, S. Palanivel, V. Ramalingam, "Classification of audio signals using SVM and RBFNN," *Expert systems with applications*, vol. 36, no. 3, pp. 6069-6075, 2009.

[27] J. Carvajal, G. Chen, H. Ogmen, "Fuzzy PID controller: Design, performance evaluation, and stability analysis," *Information sciences*, vol. 123, no. 3-4, pp. 249-270, 2000.

[28] C. B. Jabeur, H. Seddik, "Optimized neural networks-PID controller with wind rejection strategy for a Quad-Rotor," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 62-72, 2022.

[29] N. M. H. Norsahperi, S. Ahmad, S. F. Toha, M. A. A. Mutalib, "Design, Simulation and Experiment of PSO-FOPID Controller for Height Position Control of a Scissor Mechanism Platform," *FME Transactions*, vol. 50, no. 1, 2022.

[30] Z. Guan, T. Yamamoto, "Design of a reinforcement learning PID controller," *IEEJ transactions on electrical and electronic engineering*, vol. 16, no. 10, pp. 1354-1360, 2021.

[31] Y. Li, N. Sundararajan, P. Saratchandran, "Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems," *IEE Proceedings-Control Theory and Applications*, vol. 147, no. 4, pp. 476-484, 2000.

[32] D. Sbarbaro, J. P. Segovia, S. Alcozer, J. Gonzales, "Applications of radial basis network technology to process control," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 1, pp. 14-22, 2000.

[33] Y. Li, K. H. Ang, G. C. Chong, "PID control system analysis and design," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 32-41, 2006.

[34] N. Zijie, Z. Peng, Y. Cui, Z. Jun, "PID control of an omnidirectional mobile platform based on an RBF neural network controller," *Industrial Robot: the international journal of robotics research and application*, vol. 49, no. 1, pp. 65-75, 2022.

[35] M. W. Hasan, A. S. Mohammed, S. F. Noaman, "An adaptive neuro-fuzzy with nonlinear PID controller design for electric vehicles," *IFAC Journal of Systems and Control*, vol. 27, p. 100238, 2024.

NOMENCLATURE

$G_{PID}(s)$	Transfer function of the PID controller
K_p, K_i, K_d	Proportional, integral, and derivative coefficients
T_i, T_d	Integral and derivative constant time
x	The input of RBFNN
X	The input vector of RBFNN
h	The Gaussian function
H	The Gaussian function vector
c	The basis function center
C	The center vector of the basis function
b	The width of the basis function
B	The width vector of the basis function
w	The output weight
W	The output weight vector
Δw	The output weight updated by the Gradient Descent algorithm
Δb	The width of the basis function updated by the Gradient Descent algorithm
Δc	The center of the basis function updated by the Gradient Descent algorithm
$\Delta Kp, \Delta Ki, \Delta Kd$	The proportional, integral, and derivative coefficients updated by the Gradient Descent algorithm
$y_d(k)$	The desired response
$y_m(k)$	The estimated response
$y(k)$	The actual response
$e(k)$	The error between the desired and actual responses
$E(k)$	The least squared error
η	The momentum coefficient
α	The learning rate
$\Delta u(k)$	The RBF based PID controller
$\dfrac{\partial y}{\partial \Delta u}$	The Jacobian function
a_0	The initial acceleration of the cart
$a(t)$	The acceleration of the cart
v_0	The initial velocity of the cart
$v(t)$	The velocity of the cart
q	The acceleration constant
$n_j\,(j=\overline{1,3})$	The number of pulses at the j^{th} movement stage
S	The cart's travel distance
R	The radius of the pulley
p	The number of pulses to be controlled

ABBREVIATIONS

IC	Integrated circuit
FPGA	Field programmable gate array
MIMO	Multiple-input multiple-output
P	Proportional
PI	Proportional integral
PID	Proportional integral derivative
PWM	Pulse width modulation
RBFNN	Radial basis function neural network
RBF-PID	Radial basis function neural network based PID controller
VHDL	Very high speed integrated circuit hardware description language
SWD	Single-wire debug

УГРАДЊА ПИД КОНТРОЛЕРА ЗАСНОВАНОГ НА РАДИЈАЛНОЈ БАЗИ ФУНКЦИЈЕ НЕУРОНСКЕ МРЕЖЕ У МИКРОКОНТРОЛЕР ЗА КОНТРОЛУ ПОЛОЖАЈА КОЛИЦА У РЕАЛНОМ ВРЕМЕНУ

Х.Д. Нгујен, Т.Д. Ле, Т.К. Тран, Т.Х. Хујнх

Пропорционални интегрални дериват (ПИД) је класичан конвенционални регулатор и широко се користи у индустрији. Међутим, његови коефи–

цијенти се обично бирају коришћењем емпиријских метода. Ако се параметри постројења мењају на време или на њих утичу неизвесни шумови, конвенционални регулатор није стабилан због својих фиксних коефицијената. Стога, овај рад предлаже метод за уградњу ПИД контролера заснованог на радијалној бази функције неуронске мреже у микроконтролер СТМ32Ф4ВЕ за контролу положаја колица. Његови коефицијенти се процењују у реалном времену коришћењем приступа Градијентног спуштања и неуронске мреже радијалне базне функције. Два контролера која се користе за контролу колица су додала 2,5 кг носивости и 20 цм кретања. Конвенционални ПИД регулатор је направио прекорачење од 29,5 % док је предложени метод 2,2 %. Експериментални резултати показују да предложена метода може савршено да контролише положај колица са промењеном растојањем кретања и носивости колица.