

Research

A decision-making approach on control techniques for an inverted pendulum based on, neuro-fuzzy, indirect adaptive and PID controllers

Jesus de la Cruz-Alejo¹ · Hugo Beatriz-Cuellar¹ · I. C. Alcocer Guillermo¹ · Agustin Mora Ortega¹

Received: 22 January 2024 / Accepted: 26 April 2024

Published online: 07 May 2024

© The Author(s) 2024 **OPEN**

Abstract

The operation of an inverted pendulum and its respective type of control are affected by the change of the values of its internal parameters. Changes with high uncertainty result in responses with undesirable outputs. In this work, a comparison is presented for the control of an inverted pendulum to determine the operation and characteristics of three types of control systems: Neuro-Fuzzy Control (NFC), Indirect Adaptive Control (IAC) and a Proportional Integral Derivative control (PID). The study considers several indices such as stabilization time, rise time, mean square error, overshoots, convergence, computational load, error, mathematical requirements, and performance indices for control systems. To demonstrate its operation, the controls are implemented in hardware, one for the NFC and another for the IAC under an Arduino UNO platform. The results indicate that the NFC and IAC controls do not generate a transient or impulse response, only a small delay and the rise and stabilization time are minimal. While PID presents a transient response and overshoot, as well as a stabilization time to reach the steady state response.

Article Highlights

- The characteristics and operation of an NFC, IAN, and a PID were analyzed using an inverted pendulum.
- The prototype was built to carry out experimental tests with the controllers.
- For the controllers, the initial requirements, implementation results (hardware), and efficiency are described.

Keywords Arduino UNO · Artificial neural networks · Fuzzy logic control · Inverter pendulum · PID

1 Introduction

Currently, control systems are a very important element in a mechatronic process since they allow to maintain the conditions desired or required by the user. Nowadays, there are different types of control systems that are used in different types of processes [1, 2]. However, for the selection of a suitable control system, it is necessary to consider different characteristics and/or parameters such as: performance, efficiency, precision, computational load, required

✉ Jesus de la Cruz-Alejo, jdelaacruz@tese.edu.mx; Hugo Beatriz-Cuellar, 201921499@tese.edu.mx; I. C. Alcocer Guillermo, ialcocer@tese.edu.mx; Agustin Mora Ortega, amora@tese.edu.mx | ¹The School Tecnológico Nacional de México, Tecnológico de Estudios Superiores de Ecatepec, Ecatepec de Morelos, Mexico.



software or hardware, mathematical models and, most importantly, if there are large uncertainties in the process. Conventional controls are based on a rigorous mathematical model of the system. In a PID, the way to optimize the control action is to adjust variables and tune coefficients, but it is complicated due to random changes, both internal and external disturbances of the system. An alternative is to use intelligent control systems, which are a control technique that uses the knowledge of a human being in a particular control domain and attempts to imitate human intelligence to learn and make decisions to solve problems, [3]. Artificial neural networks, fuzzy logic control, generic algorithms, expert systems, and hybrid systems are the main types of intelligent control.

Also, there are other structures that combine two intelligent control systems, for example, neuro-fuzzy control, direct adaptive control, indirect adaptive control, artificial neural network and a genetic algorithm, among others. In this case, NFC and IAC were used, which use fuzzy logic control (FLC) and artificial neural networks (ANN). The FLC does not require the mathematical model of the system, experimental tests can be used to define the control strategy and linguistic values to define the characteristics of the process and the control action. ANNs do not require the mathematical model of the system, only a data set of input and output variables is used to learn the dynamics and control of the process [4, 5]. Today, an FLC is used in a wide variety of applications, for example, monitoring carbon dioxide, carbon monoxide and ambient air levels in a ventilation system [6], peak-point tracking control power a photovoltaic panel [7], optimal point tracking control power of a photovoltaic panel [7], humidity and water level control of a hydroponic system [8], regulation of intravenous administration of anti-cancer drugs [9], trajectory tracking control of a robot or manipulator [10], tracking of the maximum power point of a wind turbine [11], among others. ANNs have been used to predict the thermal conductivity of multi-walled carbon nanotubes (water nanofluid) [12], detect possible production line failures to identify and apply maintenance activities [13], predict cases and deaths caused by COVID-19 [14], analysis of environmental pollution in the European Union due to the mining industry [15], predicting diabetes in a person [16], among others. In this work, a comparative analysis of three types of controls is carried out, these are: NFC, IAC and a PID, applied to an inverted pendulum (InvPen) system. The InvPen is used, because its movement is unstable in any direction and at any time. This property provides the characteristics necessary to analyze and validate a control system. Comparing the control action of the three controllers will show the efficiency of each one. To do this, the prototype of an InvPen and the design of programming algorithms are carried out to implement an NFC and an IAC under an Arduino UNO platform. For the analysis of the performance of the NFC and the IAC, Fuzzy Logic Toolbox™ and Neural Network Toolbox™ are used, as well as experimental tests of the InvPen to analyze its functionality and behavior. The structure of this work is as follows, section two describes the characteristics of the system, the NFC and the IAC. The third section shows the implementation of the prototype. Section four shows the simulation results. Section five shows the experimental results. The sixth section conclusions are given.

2 Description of the system

Figure 1a shows the diagram of the InvPen which is mounted on a cart (InvPenCart). The movement of the InvPenCart causes instability in the InvPen, which depending on the direction of the cart causes it to fall clockwise or counterclockwise. To maintain the balance of the InvPen at 90° to the horizontal, three types of controls are used: a PID, an NFC and an IAC, and determine the response of each of them. To measure the angular position of the InvPen, a potentiometer is used as a sensor, a stepper motor is used to adjust its angular position, and a DC motor is used to move the InvPenCart. To measure the position of the InvPen, (1) is used, which is a relationship between the output voltage (0 to 5 V) of the potentiometer and the angular position (– 150° to 150°) of the InvPen, as shown in Fig. 1b). To determine the direction of movement of the InvPen, the angular movement in degrees of the potentiometer was divided into two parts. If the result of (1) gives positive degrees, the movement of the InvPen is clockwise, in the other case, the movement of the InvPen is counterclockwise. Table 1 shows the InvPen parameters used for the prototype. Figure 2 shows the prototype of the InvPen designed. To control the movement and direction of the InvPenCart, a

Fig. 1 Configuration of **a** InvPen and **b** potentiometer

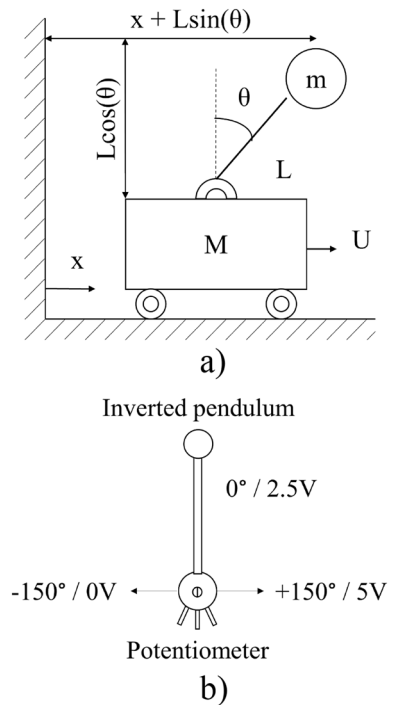
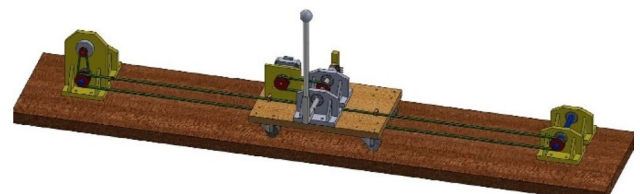


Table 1 Technical characteristics of the InvPen

M (kg)	m (kg)	L (m)	InvPenCart operating range (m)
1.0	0.12	0.30	0.8

Fig. 2 Design of the InvPen



pulley-belt-pulley transmission system is used, which has a transmission ratio of 5:1. The potentiometer, stepper motor, and the InvPen are connected by a pulley-belt-pulley transmission system with a transmission ratio of 1:1.

$$PosInvPen = \left(\frac{((VPot) * (300^\circ))}{5Vdc} \right) - 150^\circ \tag{1}$$

where VPot is the output voltage of the potentiometer, 5Vdc is the potentiometer supply voltage, and 300° is the operating range of the InvPen.

The mathematical model of the InvPen, where the friction (b) is not considered, is given by:

$$(I + mL^2) \frac{d^2\theta}{dt^2} - mgL\theta = mL \frac{d^2x}{dt^2} \tag{2}$$

Fig. 3 Block diagram of the InvPen and the PID.

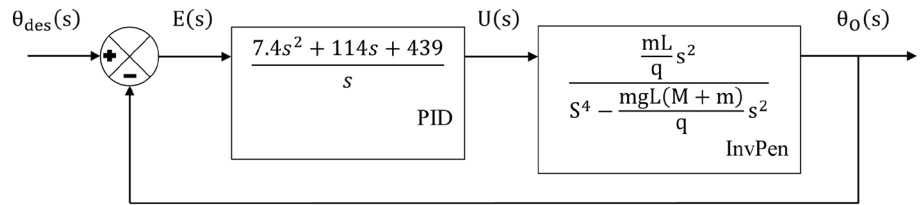
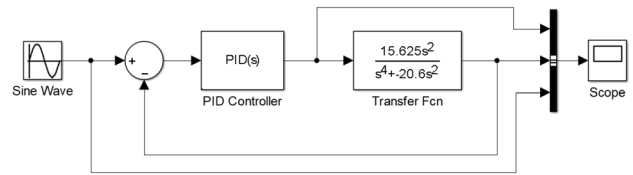


Fig. 4 Block diagram implemented in Simulink



$$(M + m) \frac{d^2x}{dt^2} + b \frac{dx}{dt} - mL \frac{d^2\theta}{dt^2} = u \tag{3}$$

Taking the Laplace transforms of (2, 3),

$$(I + mL^2) s^2\theta(s) - mgL\theta(s) = mLs^2X(s) \tag{4}$$

$$(M + m)s^2X(s) + bX(s) - mLs^2\theta(s) = U(s) \tag{5}$$

Then, the transfer function is, [17, 18]:

$$q = (M + m)(I + mL^2) - (mL)^2 \tag{6}$$

$$\frac{\theta(s)}{U(s)} = \frac{\frac{mL}{q} s^2}{s^4 - \frac{mgL(M+m)}{q} s^2} \tag{7}$$

And the solution of the mathematical model is:

$$\theta(t) = \frac{2000\sqrt{395}\sqrt{9483}\sinh\left(\frac{\sqrt{395}\sqrt{9483}}{395}t\right)}{6742413} \tag{8}$$

2.1 Proportional integral derivative control (PID)

Figure 3 shows the block diagram of the PID for the InvPen. The input signal is the error ($\theta_{des}(s)$), which is the difference between the desired position ($\theta_{des}(s)$) and the current position ($\theta_O(s)$). The output signal is the control signal ($U(s)$). The PID depends on the gain parameters: proportional K_p , integral K_i and derivative K_d . To determine these gains, a programming algorithm was performed to implement the computational optimization method [19]. The values obtained were: 7.4, 114 and 439 for K_p , K_i and K_d , respectively. Figure 4 shows the control of the InvPen using the PID. The programming algorithm to implement the computational optimization method is shown below.

```

M = 1.0; % mass of the InvPenCart
ma=0.120; % mass of the InvPen
L=0.3; % length of the InvPen
gv=9.81; % earth gravity
I=0.0075; % Moment of inertia

% Different values of "K" and "a" are tested to determine the
% best ones for the system.
K = [1.0 1.2 1.4 1.6 1.8, ..., 8.0 8.2 8.4 8.6 8.8 9.0];
a = [0.1 0.3 0.5 0.7 0.9, ..., 7.1 7.3 7.5 7.7 7.9];

q = (M + ma)*(I + ma*(L^2))-(ma*L)^2;
% g is the transfer function of the InvPen.
g = tf([(M*L)/q 0 0],[1 0 -(ma*gv*L)*(M+ma)/(q) 0 0]);

for i = 1:40;
for j = 1:40;
% gc is the PID transfer function.
% G is the closed loop transfer function.
gc = tf(K(i)*[1 2*a(j) a(j)^2],[1 0]);
G = gc*g/(1+gc*g);
y = step(G,t); % System response
m = max(y); % Maximum value of the system response

if m<1.10 % Save the best "K" and "a" values.
k=k+1;
solution(k,:) = [K(i) a(j) m]; end end end
% Response with the largest overelongation that is less than
10%
K = sortsolution(k,1); a = sortsolution(k,2);

```

2.2 Indirect adaptive control (IAC)

To tune a system, IAC uses an ANN to model the operation of the system and an ANN to generate the control signal ($u(k)$). IAC generates the signal $u(k)$, which makes the system output as similar as possible to the desired signal ($\theta_{des}(k)$). Figure 5 shows the IAC structure used to control the InvPen. The training of the ANN-M is performed using the identification error ($e_{id}(k)$), which is the difference between the system response ($\theta_{real}(k)$) and the output of the ANN-M ($\theta_{ANN}(k)$). For training the ANN-C, it is done using the tracking error ($e_{tr}(k)$), which is the difference between $\theta_{real}(k)$ and $\theta_{des}(k)$. First, the training of the ANN-M is carried out and subsequently, the training of the ANN-C [20]. ANN-M and ANN-C have an input layer with one input, an intermediate layer with 20 neurons, an output layer with one neuron, and sigmoid activation

Fig. 5 Block diagram of indirect adaptive control

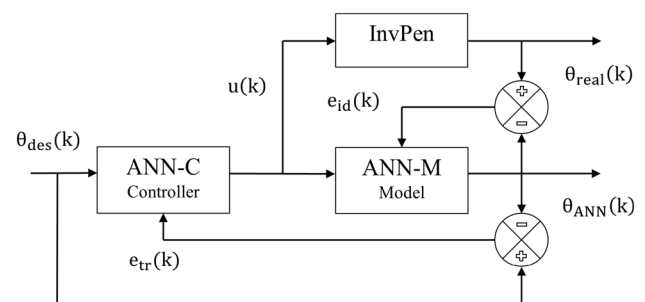
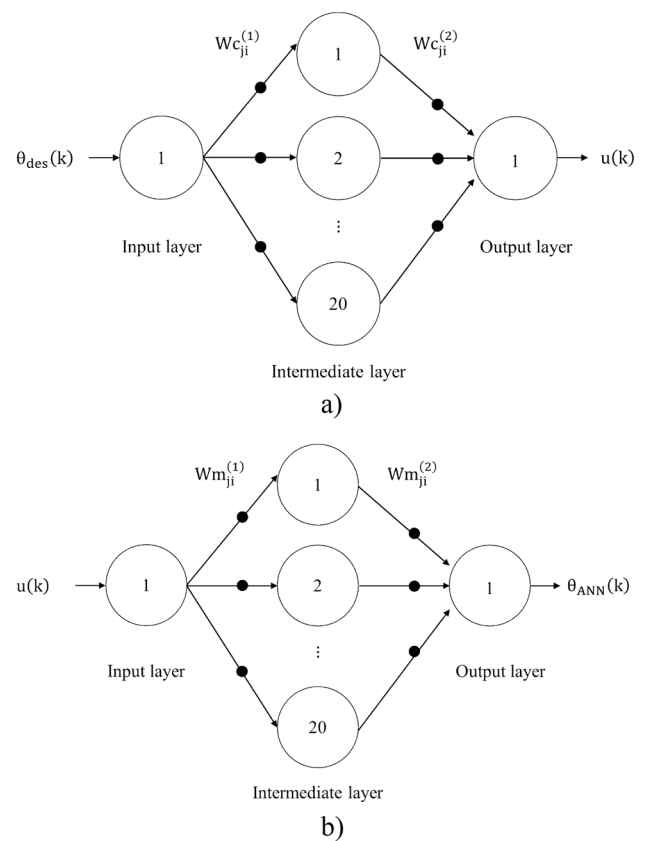


Fig. 6 Structure of **a** ANN-C and **b** ANN-M



functions. Figure 6 shows the structure of ANNs. The backpropagation algorithm is the most used method to train an ANN [21, 22]. The first stage of the method is to obtain the response of the ANN based on the value of the input variables. The second stage of the method is to update the synaptic weights ($W_{ji}^{(1)}$ and $W_{ji}^{(2)}$), for which the error must be propagated backward through the layers (L) of the ANN. The input of an artificial neuron is determined by.

$$I_j^{(L)} = \sum_{i=0}^{j-1} (W_{ji}^{(L)} * x_i) \tag{9}$$

The output of an artificial neuron is determined by.

$$O_j^{(L)} = g(I_j^{(L)}) \tag{10}$$

To apply the backpropagation algorithm to the ANN-M, the Squared Error (SE), which is the difference between $\theta_{real}(k)$ and $\theta_{ANN}(k)$, must be determined by

$$E(k) = \frac{1}{2} (\theta_{real}(k) - \theta_{ANN}(k))^2 \tag{11}$$

Applying the descending gradient to (XX) with respect to the synaptic weights ($Wm_{ji}^{(2)}$ and $Wm_{ji}^{(1)}$), we obtain:

$$A_1 = (\theta_{\text{real}}(k) - \theta_{\text{ANN}}(k)) * g'(\theta_{\text{ANN}}(k)) \quad (12)$$

$$Wm_{ji}^{(2)}(k+1) = Wm_{ji}^{(2)}(k) + \eta * A_1 * O_j^{(1)} \quad (13)$$

$$A_2 = \sum_{j=1}^{j_2} (A_1 * Wm_{ji}^{(2)}) * g'(O_j^{(2)}) \quad (14)$$

$$Wm_{ji}^{(1)}(k+1) = Wm_{ji}^{(1)}(k) + \eta * A_2 * u(k) \quad (15)$$

To apply the backpropagation algorithm to the ANN-C, the SE, which is the difference between $\theta_{\text{real}}(k)$ and the $\theta_{\text{des}}(k)$, must be determined by

$$E(k) = \frac{1}{2} (\theta_{\text{real}}(k) - \theta_{\text{des}}(k))^2 \quad (16)$$

Applying the descending gradient to (XX) with respect to the synaptic weights ($Wc_{ji}^{(2)}$ and $Wc_{ji}^{(1)}$), we obtain:

$$A_3 = \sum_{j=1}^{j_3} (A_2 * Wm_{ji}^{(1)}) * g'(O_j^{(2)}) \quad (17)$$

$$Wc_{ji}^{(2)}(k+1) = Wc_{ji}^{(2)}(k) + \eta * A_3 * O_j^{(1)} \quad (18)$$

$$A_4 = \sum_{j=1}^{j_2} (A_3 * Wc_{ji}^{(2)}) * g'(O_j^{(1)}) \quad (19)$$

$$Wc_{ji}^{(1)}(k+1) = Wc_{ji}^{(1)}(k) + \eta * A_4 * \theta_{\text{des}}(k) \quad (20)$$

The analysis of the performance of the indirect adaptive control was carried out by

$$MSE = \frac{1}{P} \sum_{k=1}^P (E(k)) \quad P : \text{number of samples} \quad (21)$$

Then, to analyze the performance of the IAC, a programming algorithm was designed on the Arduino UNO board, using only basic operations (addition, subtraction, multiplication, division and exponential). The experimental results were as follows: processing time of 4.29 ms, 48% memory, training time of 5.35 h. The programming algorithm to implement the ANN is shown below.

```

/* the sigmoidal activation function for neurons is used, dj1
is the desired output signal, X1 represents u(k) or  $\theta_{des}(k)$ ,
and "lr" is the learning rate and has a value of 0.090*/

for (k=1;k<=300;k++) { // samples to train the ANN

% Forward propagation stage
I1j1=(w1ji10*X0) + (w1ji11*X1);
I1j20=(w1ji200*X0) + (w2ji201*X1);
O1j1=1/(1+exp(-I1j1));
O1j20=1/(1+exp(-I1j20));
I2j1=(w2ji10*X0) + (w2ji120*O1j20);
O2j1=1/(1+exp(-I2j1));

% Mean Square Error
MSE=MSE+((1/2)*((dj1-O2j1)^2));

% Update of output layer synaptic weights.
err21=(dj1-O2j1)*(O2j1*(1-O2j1));
w2ji10=w2ji10+(lr*err21*X0);
w2ji201=w2ji201+(lr*err21*O1j20);

% Update of the synaptic weights of the intermediate layer
err11=(err21)*(w2ji11)*(O1j1*(1-O1j1));
err12=(err21)*(w2ji21)*(O1j2*(1-O1j2));

err119=(err21)*(w2ji191)*(O1j19*(1-O1j19));
err120=(err21)*(w2ji201)*(O1j20*(1-O1j20));

w1ji10=w1ji10+(lr*err11*X0);
w1ji11=w1ji11+(lr*err11*X1);
w1ji20=w1ji20+(lr*err12*X0);
w1ji21=w1ji21+(lr*err12*X1);

w1ji190=w1ji190+(lr*err119*X0);
w1ji191=w1ji191+(lr*err119*X1);
w1ji200=w1ji200+(lr*err120*X0);
w1ji201=w1ji201+(lr*err120*X1);

}

```

2.3 Neuro-fuzzy control (NFC)

An NFC is carried out through the union of an FLC and an ANN. The ANN training algorithm is used to tune the parameters of the fuzzy sets and the coefficients of the FLC polynomials. The configuration that was used in this work is known as the Adaptive Network-Based Fuzzy Inference System (ANFIS). For this, backpropagation algorithm and a Takagi–Sugeno type fuzzy control were used. $\theta_{des}(k)$ and a delay of the $\theta_{des}(k)$ signal were used as input variables, and $u(k)$ as the output variable. Five triangular fuzzy sets were defined for $\theta_{des}(k)$ labeled as A_1, A_2, A_3, A_4 and A_5 . For $d(\theta_{des}(k))/dt$, also five fuzzy sets labeled B_1, B_2, B_3, B_4 and B_5 . Twenty-five rules (R) were used, and the polynomials of the consequent are of first order $y = p_0 + p_1x_1 + p_2x_2$ [23–25]. Figure 7 shows the NFC structure used to control the InvPen, Fig. 8 shows the fuzzy sets of the input variables and Fig. 9 shows the structure of ANFIS. Layer 1 performs fuzzification of the input variables, this is done by.

Fig. 7 Block diagram of neural-fuzzy control

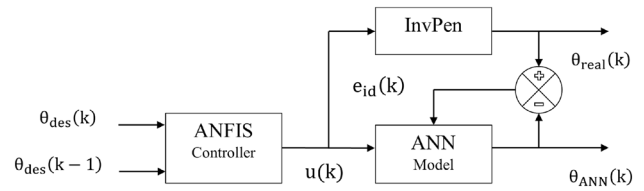


Fig. 8 Fuzzy sets of \mathbf{a} $\theta_{des}(k)$, and \mathbf{b} $\theta_{des}(k - 1)$. (x_1 and x_2 represents $\theta_{des}(k)$ and $\theta_{des}(k - 1)$, respectively.)

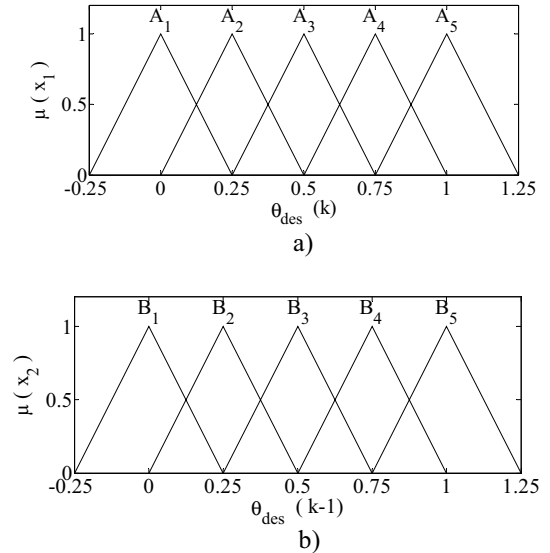
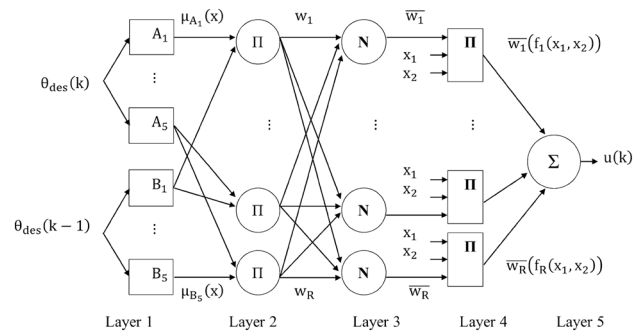


Fig. 9 ANFIS structure (x_1 and x_2 represents $\theta_{des}(k)$ and $\theta_{des}(k - 1)$, respectively.)



$$\mu_C(x_n) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a < x < b \\ \frac{c-x}{c-b} & \text{if } b < x < c \\ 0 & \text{if } x > c \end{cases} \tag{22}$$

Layer 2 determines the intensity of the rules (r), this is done by.

$$w_r = (\mu_A(x_1)) * (\mu_B(x_2)) \tag{23}$$

Layer 3 determines the intensity of a fuzzy rule with respect to the intensity of all fuzzy rules, this is done by.

$$\bar{w}_r = \frac{w_r}{w_1 + w_2 + \dots + w_R} \quad (24)$$

Layer 4 determines the consequent of the rules, this is done by.

$$y_r = \bar{w}_r * (p_0 + p_1 * x_1 + p_2 * x_2) \quad (25)$$

Layer 5 determines the global output or response of the ANFIS, this is done by.

$$Z = \sum_{r=1}^R (\bar{w}_r * y_r) \quad (26)$$

To apply the backpropagation algorithm to the ANFIS, the SE, which is the difference between $d(k)$ and $u(k)$, must be determined by

$$E(k) = \frac{1}{2}(d(k) - u(k))^2 \quad (27)$$

Applying the descending gradient to (27) with respect to the coefficients of the polynomials (p_0, p_1, p_2), and the parameters of a fuzzy set (a, b , and c), we obtain:

$$p_0(k+1) = p_0(k) - \eta(\bar{w}_r) \quad (28)$$

$$p_1(k+1) = p_1(k) - \eta(d(k)) * (\bar{w}_r) \quad (29)$$

$$p_2(k+1) = p_2(k) - \eta(\theta_{des}(k)) (d\theta_{des}(k)/dt) (\bar{w}_r) \quad (30)$$

$$W_T = \sum_{r=1}^R (w_r) \quad (31)$$

$$a(k+1) = a(k) - \eta(y_r) \left(\frac{W_T - w_r}{W_T^2} \right) (\mu_C(x_n)) \left(\frac{-b+x}{(b-a)^2} \right) \quad (32)$$

$$b(k+1) = b(k) + \eta(y_r) \left(\frac{W_T - w_r}{W_T^2} \right) (\mu_C(x_n)) \left(\frac{x-a}{(b-a)^2} \right) \quad (33)$$

$$c(k+1) = c(k) - \eta(y_r) \left(\frac{W_T - w_r}{W_T^2} \right) (\mu_C(x_n)) \left(\frac{b-x}{(c-b)^2} \right) \quad (34)$$

The analysis of the performance of neuro-fuzzy control is carried out by (21).

To analyze the performance of the NFC, a programming algorithm was designed on an Arduino UNO board, using basic operations (addition, subtraction, multiplication and division). The experimental results were as follows: a processing time of 120 ms, 43% memory. The programming algorithm used to implement NFC is shown below.

X represents $\theta_{des}(k)$, Y represents $\theta_{des}(k-1)$, lr is learning rate.

```

//////////////////// Layer 1 //////////////////////
if x>=a1 && x<=b1 % Membership function of A1.
uA1=(X-a1)/(b1-a1); end
if x>b1 && x<=c1
uA1=(c1-X)/(c1-b1); end
if y>=d1 && y<=e1 % Membership function of B1
uB1=(Y-d1)/(e1-d1); end
if y>e1 && y<=f1
uB1=(f1-Y)/(f1-e1); end
//////////////////// Layer 2 //////////////////////
w1=uA1*uB1; w2=uA1*uB2; ... w25=uA5*uB5;
//////////////////// Layer 3 //////////////////////
Ewk=w1 + w2 + ... + w25;
w1nor=w1/Ewk; w2nor=w2/Ewk; ... w25nor=w25/Ewk;
//////////////////// Layer 4 //////////////////////
y1=m10+((m11)*(x))+((m12)*(y));
y2=m20+((m21)*(x))+((m22)*(y));
y25=m250+((m251)*(x))+((m252)*(y));
//////////////////// Layer 5 //////////////////////
O1=w1nor*y1; O2=w2nor*y2; ... O25=w25nor*y25;
Ey=O1 + O2 + ... + O24 + O25;
e=yd-Ey;

// Updating the coefficients of the polynomials
m10=m10+lr*((e)*(1)*(1)*(w1nor));
m11=m11+lr*((e)*(1)*(X)*(w1nor));
m12=m12+lr*((e)*(1)*(Y)*(w1nor));

// Updating the parameters of a fuzzy set
WT=w1 + w2 + ... +w24 + w25;
a1=a1+lr*((e)*(1)*(y1)*((WT-w1)/(WT^2))*(uB1)*((-
b1+X)/((b1-a1)^2)));
b1=b1+lr*((e)*(1)*(y1)*((WT-w1)/(WT^2))*(uB1)*((X-
a1)/((b1-a1)^2)*(-1));
c1=c1+lr*((e)*(1)*(y1)*((WT-w1)/(WT^2))*(uB1)*((-
b1+X)/((c1-b1)^2)));

```

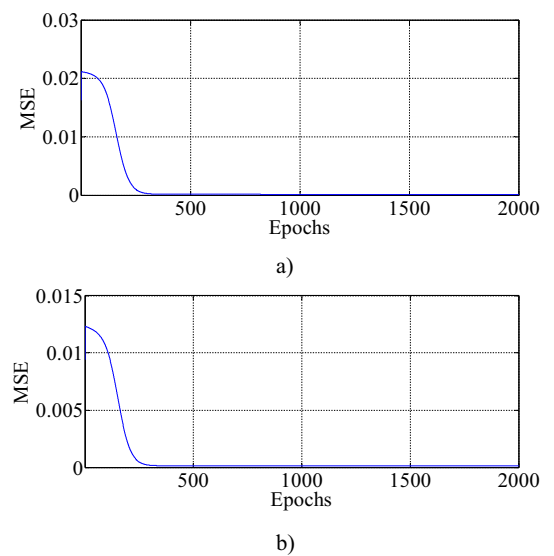
3 Implementation of the ip prototype

Figure 10a shows the electronic system of the InvPen, which is composed of an Arduino UNO board, a 10K Ω potentiometer, a stepper motor and an L298N module [26, 27]. Furthermore, Fig. 10b and c show the InvPen prototype. The position of the InvPen was determined using the potentiometer, which was connected to an analog input of the Arduino UNO board. To determine the control action of the InvPen, the Arduino UNO board performs the NFC or IAC process, the output of which is used to adjust the position of the InvPen by sending control signals to the L298 module, which provides the energy to move the motor.

4 Simulation results

For the analysis of NFC, the Fuzzy Logic Toolbox was used and for IAC, the Neural Network Toolbox was used [28, 29]. To analyze the two types of control, it was carried out by obtaining the MSE, which indicates the precision of the system modeling by quantifying the difference between the desired results and the actual results. For this reason, the MSE is

Fig. 10 MPN training result, **a** comparison of the reference signal and the output of the MPN, and **b** MSE



used to train in IAC and ANFIS. The experimental results obtained were: an MSE of 2.15101×10^{-4} for the ANFIS, and an MSE of 1.4608×10^{-4} for the IAC. Based on the above, the InvPen control presented a high degree of precision, since small MSE values indicate that the process modeling has less error. In this case, ANFIS and IAC managed to learn the dynamics of the InvPen. Figure 11a shows the reduction of MSE during ANFIS training and Fig. 11b shows the reduction of MSE during IAC training.

5 Experimental results

For performance testing of the PID, IAC and NFC, a step signal, a ramp signal, and a sine signal were used. Figure 12 shows the response of the three controllers to a step signal. In this case, the PID showed an overshoot of 10% and a settling time of 0.3 s. While the IAC and the NFC did not present an overshoot and the settling time is shorter. The overshoot that is observed is the maximum point that exceeds the desired level. The settling time is the time necessary to stabilize the system. A transient response can be observed, which causes the system to converge slowly. Figure 13 shows the response of the three controllers to a ramp signal. Figure 14 shows the response of the three controllers to a sinusoidal signal. As can be seen, none of the reference signals generated an overshoot in the InvPen

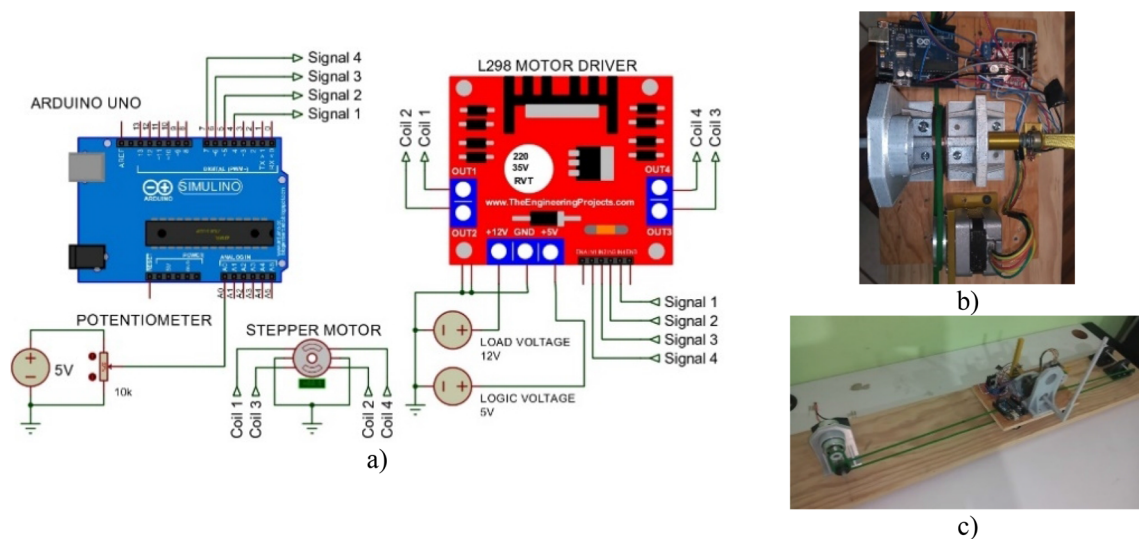


Fig. 11 IP, **a** electronic diagram, **b** implementation of the electronic diagram, and **c** prototype

Fig. 12 InvPen response to a step signal

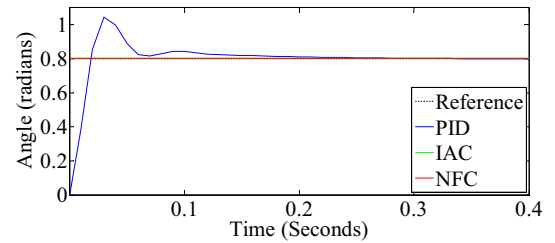


Fig. 13 InvPen response to a ramp signal

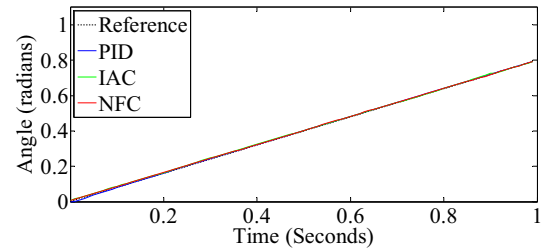


Fig. 14 InvPen response to a sinusoidal signal

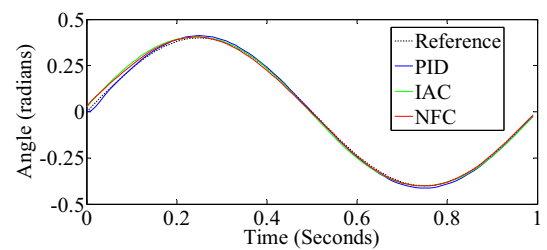


Fig. 15 PID control signals

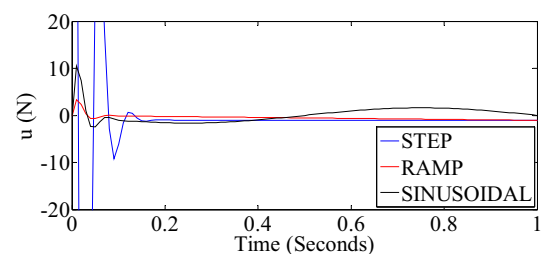
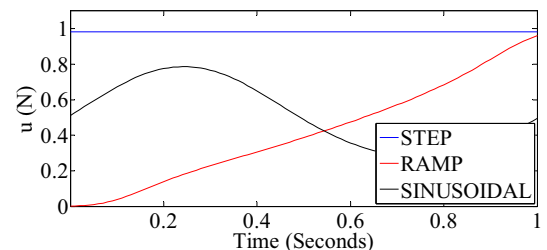


Fig. 16 IAC control signals



response. Figure 15 shows the PID control signals for the reference signals. Figure 16 shows the IAC control signals for the reference signals. Figure 17 shows the NFC control signals for the reference signals. Additionally, Integral Square Error (ISE), Integral Absolute Error (IAE), Integral time Squared Error (ITSE), and Integral time Absolute Error (ITAE) were determined for the three controllers. These parameters are used to analyze the operation of a control system. Tables 2, 3 and 4 show the ISE, IAE, ITSE and ITAE values of the controllers for the different reference inputs.

On the other hand, the analysis of the robustness of the controllers was carried out by analyzing the response of the system to a disturbance. Figure 18 shows the step signal with a disturbance, Fig. 19 shows the ramp signal

Fig. 17 NFC control signals

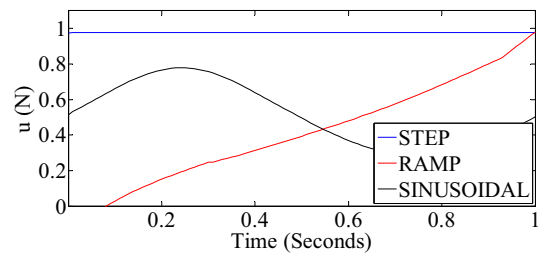


Table 2 PID performance indices

	ISE	IAE	ITSE	ITAE
Step	0.0061	0.0185	6.786e-5	0.0009
Ramp	6.508e-6	0.0023	2.961e-6	0.0011
Sinusoidal	8.418e-5	0.0079	3.843e-5	0.0038

Table 3 IAC performance indices

	ISE	IAE	ITSE	ITAE
Step	2.471e-6	0.0015	1.223e-6	0.0007
Ramp	1.906e-6	0.0010	1.263e-6	0.0005
Sinusoidal	4.683e-5	0.0049	2.545e-5	0.0026

Table 4 NFC performance indices

	ISE	IAE	ITSE	ITAE
Step	1.368e-5	0.0036	6.781e-6	0.0080
Ramp	1.224e-6	0.0007	6.906e-7	0.0004
Sinusoidal	3.210e-5	0.0043	1.707e-5	0.0024

Fig. 18 System response to a step signal with a disturbance

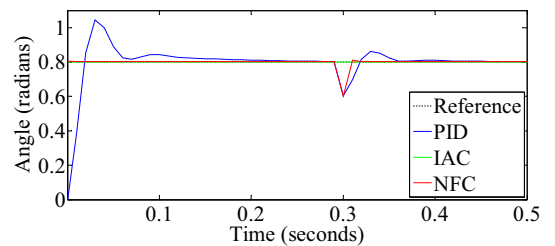


Fig. 19 System response to a ramp signal with a disturbance

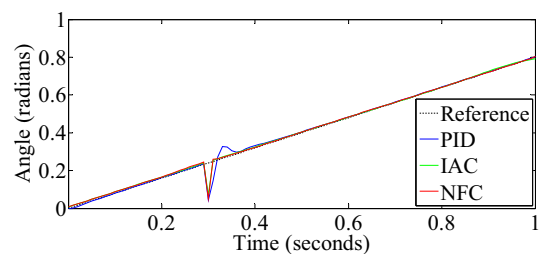


Fig. 20 System response to a sinusoidal signal with a disturbance

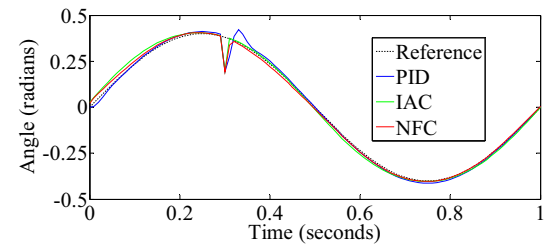


Table 5 Advantages and disadvantages of the NFC, the IAC, and the PID

	Advantages	Disadvantages
Indirect Adaptive	It does not require the mathematical model A data set can be used to learn the dynamics of the system It can be used for linear and nonlinear systems	A large amount of data is required Knowledge of the system is required to generate the response
Neuro-Fuzzy	It does not require the mathematical model A data set can be used to learn the dynamics of the system It can be used for linear and nonlinear systems	Different tests need to be done to tune the controller The training process is slow
PID	Most widely used controller in the industry There are different methods to adjust the controller parameters	It requires the mathematical model It is mainly used for linear systems Noise can upset the controller Oscillations are generated in the system response

Table 6 Performance indices

	ISE	ITSE	IAE	ITAE
FLC	39	76.17	11.57	34.99
IAC	4.683e-5	2.545e-5	0.0049	0.0026
NFC	3.210e-5	1.707e-5	0.0043	0.0024

with a disturbance, and Fig. 20 shows the sinusoidal signal with a disturbance. As can be seen in the figures, the adjustment of the InvPen with the PID required a longer settling time than the settling time of the NFC or the IAC. In addition, impulses and oscillations were generated in the system response. On the contrary, the adjustment of the InvPen with the NFC or the IAC did not generate impulses and oscillations in the system response. Furthermore, the NFC and IAC had a shorter settling time than the settling time of the PID. Table 5 describes the main advantages and disadvantages of the responses of the three types of control, neuro-fuzzy, indirect adaptive and PID.

5.1 Comparación entre los controladores propuestos y otros trabajos

In [30], the InvPen was controlled with a fuzzy logic control (FLC) optimized by the maximum entropy principle and a genetic algorithm. To implement the control system, two encoders, two data acquisition cards and Simulink were used, which was used to monitor the system response, and generate the control signal and the reference signal for the InvPen. Table 6 shows the performance indices of FLC, IAC, and NFC. As can be seen, the proposed controllers (IAC and NFC) have better performance than the FLC.

In [31], the InvPen was controlled with a PID along with a radial-based artificial neural network. The control system was analyzed with three reference signals (step signal, square signal and sine signal). In this sense, the response of the InvPen to the different reference signals had overshoots. Additionally, the system response in the presence of a disturbance had overshoots. In comparison, the proposed controllers (NFC and IAC) did not allow overshoots to be generated in the system response. Likewise, the IAC and the NFC prevented a disturbance from generating overshoots in the system response.

In [32], the InvPen was controlled with a PID control, whose gain values are determined by the pole placement method. The response of the InvPen with this controller had overshoots and oscillations. In comparison, the IAC and NFC prevented

the generation of overshoots and oscillations, indicating that the settling time with the IAC and NFC is shorter than the settling time with the PID.

6 Conclusions

In this work, it has been demonstrated through the design and implementation of three types of control applied to an inverted pendulum, that intelligent neuro-fuzzy and indirect adaptive controls are more robust and have greater precision than PID control. The two intelligent controls kept the InvPen in balance and with greater precision. For this, the three types of control were designed and implemented in hardware for their characterization. Both neuro-fuzzy and indirect adaptive programming algorithms only use basic operations (addition, subtraction, multiplication, and division) and two control statements (IF and FOR). The tuning and its behavior were carried out based on the input and output variables with respect to various indices such as settling time, rise time, mean square error, overshoots, convergence, and performance indices for control systems. Both experimental and simulation results demonstrate that intelligent control based on neuro-fuzzy and indirect adaptive provide fast response times with practically no overshoots, which is adequate to keep the IP in balance with great precision.

Author contribution All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Jesus de la Cruz-Alejo, Hugo Beatriz-Cuellar, Irving Alcocer Guillermo and Agustin Mora Ortega. The first draft of the manuscript was written by Jesus de la Cruz Alejo and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors have not disclosed any funding.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests All authors declare that they have no conflicts of interest and there is no conflict of interest in the manuscript. Also, the authors have no relevant financial interests or private connections, and they have no interests related to the material in the manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Jin X, et al. Simulation of hydraulic transplanting robot control system based on fuzzy PID controller. *Measurement*. 2020;164:108023.
2. Basile T, Marsico AD, Perniola R. Use of artificial neural networks and NIR spectroscopy for non-destructive grape texture prediction. *Foods*. 2022;11(3):281.
3. Liu J. *Intelligent control design and Matlab simulation*. Singapore: Springer; 2018.
4. Lin HR, Cao BY, Liao YZ. *Fuzzy sets theory preliminary*. Cham: Springer International Publishing; 2018.
5. Hopgood AA. *Intelligent systems for engineers and scientists: a practical guide to artificial intelligence*. Boca Raton: CRC Press; 2021.
6. Sharma S, Obaid AJ. Mathematical modelling, analysis and design of fuzzy logic controller for the control of ventilation systems using MATLAB fuzzy logic toolbox. *J Interdiscip Math*. 2020;23(4):843–9.
7. Hassan T-U, et al. A novel algorithm for MPPT of an isolated PV system using push pull converter with fuzzy logic controller. *Energies*. 2020;13(15):4007.
8. Chen CH, Jeng SY, Lin CJ. Fuzzy logic controller for automating electrical conductivity and pH in hydroponic cultivation. *Appl Sci*. 2022;12(1):405.
9. Karar ME, El-Garawany AH, El-Brawany M. Optimal adaptive intuitionistic fuzzy logic control of anti-cancer drug delivery systems. *Biomed Sig Process Control*. 2020;58:101861.
10. Yu X, et al. Fuzzy logic control of an uncertain manipulator with full-state constraints and disturbance observer. *IEEE Access*. 2020;8:24284–95.

11. Liu Q, et al. Adaptive bias RBF neural network control for a robotic manipulator. *Neurocomputing*. 2021;447:213–23.
12. Batti H, Ben Jabeur C, Seddik H. Autonomous smart robot for path predicting and finding in maze based on fuzzy and neuro-Fuzzy approaches. *Asian J Control*. 2021;23(1):3–12.
13. Kang Z, Catal C, Tekinerdogan B. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. *Sensors*. 2021;21(3):932.
14. Kuvvetli Y, et al. A predictive analytics model for COVID-19 pandemic using artificial neural networks. *Decis Anal J*. 2021;1:100007.
15. Brodny J, Tutak M. The use of artificial neural networks to analyze greenhouse gas and air pollutant emissions from the mining and quarrying sector in the European Union. *Energies*. 2020;13(8):1925.
16. Bukhari MM, et al. An improved artificial neural network model for effective diabetes prediction. *Complexity*. 2021;2021:1–10.
17. Ogata K. *Ingeniería de Control Moderna, 4ª*. 1993.
18. Ogata K, Sanchez GLP. *Dinámica de sistemas*. Prentice-Hall Hispanoamericana, 1987.
19. Ogata K. *Ingeniería de control moderna*. Pearson Educación, 2003.
20. Boutalis Y, et al. *System identification and adaptive control. Theory and Applications of the neurofuzzy and fuzzy cognitive network models*, 2014.
21. Da Silva IN, et al. *Artificial neural network architectures and training processes*. Cham: Springer International Publishing; 2017.
22. Rosa JPS, et al. *Using artificial neural networks for analog integrated circuit design automation*. Berlin: Springer; 2020.
23. Matía F, Marichal GN, Jiménez E (ed.). *Fuzzy modeling and control: theory and applications*. 2014.
24. Hooda DS, Raich V. *Fuzzy Logic Models and Fuzzy Control*. En *An Introduction*. Alpha Science International, 2017.
25. Gottwald S. *Fuzzy sets and fuzzy logic: The foundations of application—from a mathematical point of view*. Springer-Verlag, 2013.
26. Artero ÓT. *ARDUINO. Curso práctico de formación*. RC libros, 2013.
27. Peña C. *Arduino IDE: Domina la programación y controla la placa*. RedUsers, 2020.
28. Knight A. *Basics of MATLAB and Beyond*. CRC press, 2019. <https://doi.org/10.1201/9780429186882>
29. Sivanandam SN, et al. *Introduction to fuzzy logic using MATLAB*. Berlin: Springer; 2007.
30. Jain A, et al. Real-time swing-up control of non-linear inverted pendulum using Lyapunov based optimized fuzzy logic control. *IEEE Access*. 2021;9:50715–26.
31. Gao H, Li X, Gao C, Wu J. Neural network supervision control strategy for inverted pendulum tracking control. *Discret Dyn Nat Soc*. 2021;2021:1–14.
32. Shreedharan S, Ravikumar V, Mahadevan SK. Design and control of real-time inverted pendulum system with force-voltage parameter correlation. *Int J Dynam Control*. 2021;9(4):1672–80.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.