Received: September 5, 2023. Revised: September 28, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

PID Neural Controller Design for Nonlinear Inverted Pendulum System

Zaid S. Mohsen[1]* **Mohamed Jasim Mohamed[1]**

[1]*Control and System Engineering Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: cse.20.20@grad.uotechnology.edu.iq

Abstract: The inverted pendulum (IP) system, is a highly coupled, complex, nonlinear system in which the performance of the system is adversely affected by parameter uncertainty and outside disturbances. Therefore, these complications must be managed by the controllers created for such systems. The primary objective of this work is to develop four control structures, including integer order proportional integral derivative neural network controllers for inverted pendulums that deal with trajectory tracking issues. Proportional-integral-derivative neural network structure1 (PIDNNS1), proportional-integral-derivative neural network structure2 (PIDNNS2), proportional-integral-derivative neural network structure3 (PIDNNS3), and proportional-integral-derivative neural network structure4 (PIDNNS4) are the controller structures for inverted pendulum (IP) system . The ant colony optimization (ACO) is a metaheuristic optimization method that is offered to optimize. the controllers' settings while minimizing the cost function. The proposed controllers' resilience to outside disturbances and parameter uncertainty is also tested. The results using MATLAB code demonstrate that the PIDNNS4 controller, which best has a reduced cost function equal to (1.177494), (1.273627), (1.209761) for trajectory tracking, parameters uncertainty, and disturbances rejection for the inverted pendulum (IP) system. and the best controller for stabilization with a low-cost function is the PIDNNS1 controller (1.280839).

Keywords: Inverted pendulum (IP) system, PID controller, Recurrent neural network (RNN), Ant colony optimization (ACO).

1. Introduction

Inverted pendulum (IP), system with two degrees of freedom (angular pendulum movement and cart location) and a single control input, orthodox control is difficult to implement. Given how complex the issue is, one must choose a dynamic system to analyze it as a model and develop a law to govern the system [1].

A cart, an inverted pendulum, an actuator unit, and a straight rail make up the inverted pendulum system. This cart has unrestricted left and right movement. The inverted pendulum, one of the fundamental dynamics and control systems, is a subject of control engineering because of its intrinsic instability and nonlinearity [2].

Various controllers have been suggested for the cart inverted pendulum by several authors. On an inverted pendulum, the principles of a fuzzy logic-based controller are implemented although the actuator dynamics and parametric uncertainty of the system could not be captured by mathematical modeling the robustness features of the fuzzy logic controller could carry out the swing-up motion [3].

In the work, a perceptual control theory (PCT) controller for a two-wheeled inverted pendulum is directly compared for the first time with a traditional control approach, LQR. the PCT controller's performance is equivalent to that of the linear–quadratic regulator (LQR) controller and is superior at disturbance rejection [4].

The primary goal of the study is to present a comparison of fuzzy logic controller (FLC), radial basis function neural network (RBF), and integral sliding mode control (ISMC) adjusted with whale optimizer algorithm (WOA) for the control of the angle position and velocity of the inverted pendulum system. According to comparison data ISMC-WOA performs better than other approaches in terms of

settling time and overshoot [5].

The primary goal of this effort is to create an efficient control system for an inverted pendulum system that can be used as both a mathematical model and a real-world prototype. The control strategy involves a logical combination of two traditional PID controllers with an appropriate metaheuristic optimization method, such as PSO (particle swarm optimization). The control technique suggested in this work is feasible and successful, as shown by simulation results on the simulink model and experiment results on the real-world model [6].

Asynchronous advantage actor-critic (A3C) algorithm-based adaptive PID controller. First off, by taking advantage of the A3C structure's multi-thread asynchronous learning properties, the controller can simultaneously train numerous agents of actor-critic structures. Second, in order to find the best parameter-tuning strategy in continuous action space, each agent employs a multilayer neural network to approach the strategy function and value function. The simulation results show that, in comparison to traditional controllers, our suggested controller can achieve fast convergence and great adaptability [7].

An optimized fuzzy logic controller is employed for the swing-up control and stabilization of a rigidly connected twin-arm inverted pendulum system.In comparison to previous state-of-the-art controllers, the experimental results demonstrate an improvement in the transient and steady-state response of the controlled system [8].

This work focuses on the construction of robust fractional-order PID (PID) controllers for tracking and stabilization of inverted pendulum (IP) systems. an optimization using a particle swarm (PSO). To demonstrate the resilience of the suggested controllers, simulation results are also obtained by including disturbances in the model [9].

The nature of mechanical systems, such as an inverted pendulum (IP), involve various factors that can impact controller design, including nonlinear differential equations, gravitational forces, impacts, complexity, friction models, noise, parameter changes, external disturbances, and uncertainties. Despite this complexity, PID controllers have been widely used due to their simplicity, good performance, ease of installation and maintenance, and competitive cost-to-performance ratio.However, controlling nonlinear systems remains a challenging task. Artificial intelligence (AI) methods, particularly approaches based on recurrent neural network (RNN), offer promising solutions. RNN, with their dynamic nature and computational power, can accurately approximate any nonlinear system. Their feedback loop enables effective control. Traditional controller tuning processes are often difficult and time-consuming, and they may not account for system nonlinearities or uncertainties. In contrast, intelligent optimization methods, such as ant colony optimization, have demonstrated success in achieving optimal tuning for PID controllers. Combining neural networks with optimization techniques enhances control technology's robustness.

The main focus of this work is to enhance the resilience and adaptability of the controller design by combining four structures of the PID neural network controller (PIDNN) and utilizing the ant colony optimization (ACO) approach to obtain optimal parameter values for the controller. The contributions of this work can be summarized as follows:

1) Integration of PIDNN structures: The study combines four different structures of the PID neural network controller, which enables a more potent and adaptable controller design.
2) Resilience improvement: By utilizing the PIDNN structures, the controller becomes more resilient to complex mechanical system dynamics, including nonlinearities, uncertainties, and disturbances. The neural network component allows for better approximation and adaptation to varying system conditions, leading to improved control performance.
3) When compared with [9] , the suggested PIDNN controllers' results are better than or converge to the best values obtained from the existing controllers.
4) Adaptive parameter tuning: The ACO approach is employed to determine the optimal values for the PIDNN controller's parameters.

The remaining sections of this work are organized as follows:

Section 2: This section provides an introduction to the dynamic model of an inverted pendulum (IP) system. Section 3: The proposed ant colony optimization (ACO) approach is described in more detail. Section 4: This section presents the design of four inverted pendulum PID neural network (PIDNN) controller structures, namely controller structures PIDNNS1, PIDNNS2, PIDNNS3, and PIDNNS4. Section 5: Simulation results of the suggested controllers are provided in this section. Section 6: The main conclusions.

2. The dynamic model of the nonlinear inverted pendulum system

The pendulum control system has been regarded as a model application for nonlinear system control single-input-multiple-output (SIMO). A well-known control problem that is applied in colleges all over the

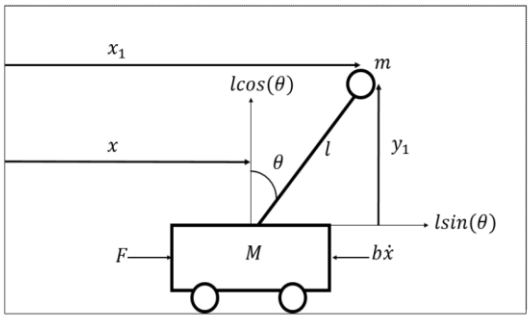


Figure. 1 The structure of the inverted pendulum

world is the inverted pendulum system. Due to its strong non-linearities and lack of stability, it is a good approach to evaluating prototype controllers. The system comprises a cart that may freely move in the x direction and an inverted pole that is hinged on it [10, 11].

The inverted pendulum system is a common multi-variable, nonlinear, and naturally unstable system. Its control techniques are widely used in the military, aerospace, robotics, and industrial process areas [12].

Based on the study of inverted pendulum control systems for industrial applications. A common vehicle today is the Segway Transporter, also referred to as a unicycle. It is a vehicle made from an inverted pendulum system and a balancing robot. This system has an unknown time-varying control coefficient and is a nonlinear uncertain system. The axle of a wheel or pair of wheels that are propelled by an electric motor serves as the pivot of the pendulum in a Segway Transporter. The controller that was created to dynamically balance the pendulum stabilizes Segway movement [13].

A conventional inverted pendulum system is built on a motorized cart connected to a pendulum, also referred to as a stick balancer. This system's control input is the force F that can move the cart horizontally, and its outputs are the pendulum's angular position and the cart's horizontal position. The euler-lagrange is used to analyze the inverted pendulum forces. The euler-lagrange equation is necessary to complete our understanding of energy. There are two primary types of energy: kinetic energy and potential energy [14].

Step 1: From the free-body diagram, the generalized coordinate is described as follows:

$$x_1 = x + l\,sin(\theta) \tag{1}$$

$$y_1 = l\cos(\theta) \tag{2}$$

Step 2: the velocity components are:

$$\dot{x}_1 = \dot{x} + l\dot{\theta}\cos(\theta) \tag{3}$$

$$\dot{y}_1 = -l\dot{\theta}\sin(\theta) \tag{4}$$

Step 3: the kinetic energy of the system is the sum of the kinetic energies of each mass, The kinetic energy general equation is:

$$K = \frac{1}{2}mV^2 \tag{5}$$

V is the velocity of a body. m is the mass of the body.ω is the angular velocity of the body. I is the moment of inertia of a body. The average velocity is:

$$V^2 = \dot{x}^2 + \dot{y}^2 \tag{6}$$

$$\omega^2 = \dot{\theta}^2 \tag{7}$$

The kinetic energy of the cart is:

$$T_1 = \frac{1}{2}M\dot{x}^2 \tag{8}$$

The kinetic energy of the pole is:

$$T_2 = \frac{1}{2}m\left(\dot{x}_1{}^2 + \dot{y}_1{}^2\right) \tag{9}$$

The total kinetic energy of the system is :

$$T = T_1 + T_2 \tag{10}$$

$$T = \frac{1}{2}(M+m)\dot{x}^2 + ml\,\dot{x}\,\dot{\theta}cos\theta + \frac{1}{2}ml^2\dot{\theta}^2 \tag{11}$$

Step 4: the potential energy of the system is:

$$V = mglcos(\theta) \tag{12}$$

Step 5: the lagrange function is:

$$L = T - V \tag{13}$$

$$L = \frac{1}{2}(M+m)\dot{x}^2 + ml\,\dot{x}\,\dot{\theta}cos\theta + \frac{1}{2}ml^2\dot{\theta}^2 - mglcos\theta \tag{14}$$

Step 6: Finally, the two euler-lagrange equations are

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = u \tag{15}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{16}$$

Step 7: Finding the euler-lagrange Eq. (15):

$$\frac{\partial L}{\partial \dot{x}} = (M+m)\dot{x} + ml\dot{\theta}cos\theta \quad (17)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = (M+m)\ddot{x} + ml\left(-\dot{\theta}^2 sin\theta + \ddot{\theta}cos\theta\right) \quad (18)$$

$$\frac{\partial L}{\partial x} = 0 \quad (19)$$

$$(M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2 sin\theta = u \quad (20)$$

$$\ddot{x} = \frac{ml\dot{\theta}^2 \sin(\theta) - ml\ddot{\theta}\cos(\theta) + u}{(M+m)} \quad (21)$$

Step 8: Finding the euler-lagrange equation (16):

$$\frac{\partial L}{\partial \dot{\theta}} = ml\dot{x}\cos\theta + ml^2\dot{\theta} \quad (22)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = -ml\dot{x}\,\dot{\theta}sin\,\theta + ml\ddot{x}\cos\theta + ml^2\ddot{\theta} \quad (23)$$

$$\frac{\partial L}{\partial \theta} = -ml\dot{x}\dot{\theta}sin\theta + mglsin\theta \quad (24)$$

$$ml^2\ddot{\theta} + m\ddot{x}l\cos\theta - mgl\sin\theta = 0 \quad (25)$$

$$\ddot{x}\cos\theta - g\sin\theta + l\ddot{\theta} = 0 \quad (26)$$

$$\ddot{\theta} = \frac{g\sin(\theta) - \ddot{x}\cos(\theta)}{l} \quad (27)$$

Step 9: Substituting Eq. (21) in Eq. (26) yields,

$$\ddot{\theta} = \frac{(M+m)gsin\theta - ml\dot{\theta}^2 sin\theta cos\theta - u\,cos\theta}{Ml + ml\sin^2\theta} \quad (28)$$

Step 10: Substituting Eq. (27) in Eq. (20) yields,

$$\ddot{x} = \frac{ml\dot{\theta}^2 sin\theta - mgsin\theta cos\theta + u}{M + m\sin^2\theta} \quad (29)$$

Thus the state space equations are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{(M+m)gsinx_1 - mlx_2^2 sinx_1 cosx_1 - u\,cosx_1}{Ml + ml\sin^2 x_1} + d_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{mlx_2^2 sinx_1 - mgsinx_1 cosx_1 + u}{M + m\sin^2 x_1} + d_2 \end{aligned} \quad (30)$$

3. Ant colony optimization (ACO)

The intelligence of the swarm and the behavior of ants in search of food inspired the ant colony algorithm, which is a search technique for the best solutions. Because the ants walk at random. The ants add pheromone to the short paths that go to a good food source and the pheromone on the long paths declines or evaporates. the method is built in three stages: first, pheromone representation, second, probabilistic solution building, and third, pheromone updating. Overall, ACO offers interactive and adaptive capabilities, distributability, the ability to handle large-scale problems, and solution diversity. These characteristics make it a valuable optimization technique in various domains [15].

Table 1. The parameter of the inverted pendulum

parameters	Nominal value
Mass of cart, M	1[kg]
Mass of pendulum, m	0.1[kg]
Length of a bar, l	0.3[m]
Standard gravity, g	9.81[m/s^2]
the angle of the pole	x_1
the angular velocity of the pole	x_2
the position of the cart	x_3
the velocity of the cart	x_4
the control signal	u
the disturbance	d_1, d_2

Table 2. Structure of solutions kept by ACOR

s_1^1	s_1^2	.....	s_1^i	.....	s_1^n	$f(s_1)$	w_1
s_2^1	s_2^2	.....	s_2^i	.....	s_2^n	$f(s_2)$	w_2
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
s_l^1	s_l^2	.....	s_l^i	.....	s_l^n	$f(s_l)$	w_l
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
s_k^1	s_k^2	.....	s_k^i	.....	s_k^n	$f(s_k)$	w_k

3.1 Pheromone representation

The structure of solutions is kept by ACO Table 2.

The structure is created with random solutions before the algorithm begins. The ants build new solutions during the running of the algorithm, with the bad solutions being deleted and the good solutions being saved in the structure. The size of the structure is (n×k), which is where solutions are kept. (n) is the dimensionality of the problem, (k) is the dimensionality of the solution [16].

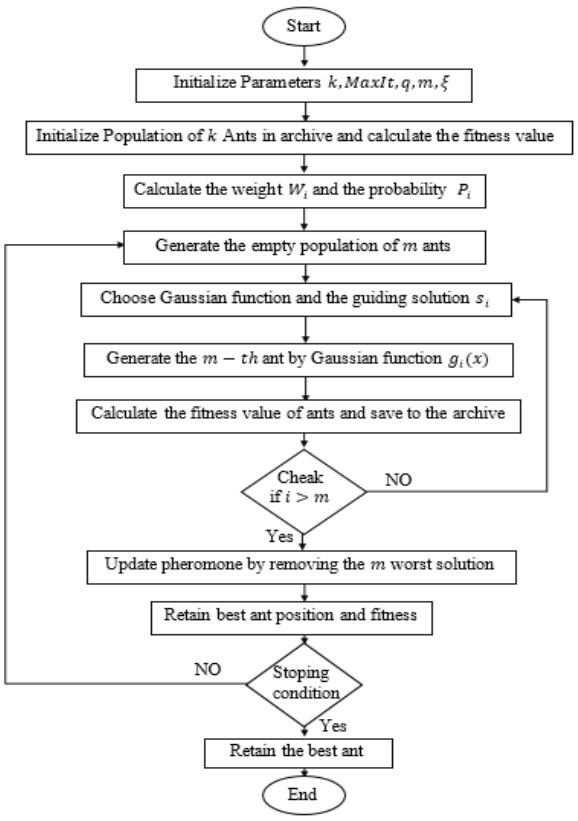


Figure. 2 Flowchart of ACO algorithm

3.2 Probability density function (PDF)

The Gaussian function is used, which is one of the most common functions to represent the probability density function because it is easy to take samples, flexibility, and non-linear properties allow for more flexible control of weights. A Gaussian kernel is indicated by the symbol $G^i(x)$

$$G^i(x) = \sum_{l=1}^{k} \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{\left(x-\mu_l^i\right)^2}{2\sigma_l^i}} \quad (31)$$

3.3 Flowchart of ACO algorithm

The algorithm ACO implementation steps [16] and the flowchart in Fig. 4 are as follows:

4. PID neural network

PID (proportional-integral-derivative) controller is a widely used feedback control algorithm in engineering applications. It is a simple but effective way to control a system by adjusting a control variable based on the error between the desired set-point and the actual process variable [17].

Using a neural network in conjunction with a PID controller has several benefits, including increased control performance, resilience, adaptability, reduced tuning effort, and scalability in complicated control systems. It is important to keep in mind that creating and training a PID neural network can be difficult and calls for knowledge of both control theory and neural networks. Additionally, the quantity and quality of the training data may affect the network's performance, and the complexity of the network may raise the processing needs of the control system [18-20]. Therefore the IP system is proposed to be controlled by four structures of PIDNN controllers.

4.1 PIDNNS1 controller

The PIDNNS1 controller is shown in Fig. 3. In this case, reference compensation technique (RCT) corrects the system being controlled by the stated controllers by closing an additional outside loop. [21]. Therefore using the following equations:

Layer 1:

$$I_i(n) = [e_\theta(t), e_\theta(t-1), e_\theta(t-2), e_x(t), e_x(t-1), e_x(t-2)] \quad (32)$$

Layer 2:

$$H_j(n) = f_j\left(\sum_i W_{ji} \cdot I_i(n) + b_j\right) \quad (33)$$

$$f_j\left(H_j(n)\right) = \frac{1-\exp\left(-H_j(n)\right)}{1+\exp\left(-H_j(n)\right)} \quad (34)$$

Layer 3:

$$O_k(n) = \sum_j V_{kj} \cdot H_j(n) + b_k \quad (35)$$

The control input for a pendulum angle:

$$u_\theta = k_{P\theta} e_\theta(t) + k_{i\theta} \int e_\theta(t)dt + k_{d\theta} \dot{e}_\theta(t) + \mathrm{O}_\theta \quad (36)$$

$$\mathrm{O}_\theta = k_{p\theta} O_1 + k_{d\theta} O_2 + k_{i\theta} O_3 \quad (37)$$

The control input for the cart position:

$$u_x = k_{px} e_x(t) + k_{ix} \int e_x(t)dt + k_{dx} \dot{e}_x(t) + \mathrm{O}_x \quad (38)$$

$$\mathrm{O}_x = k_{ix} O_4 + k_{dx} O_5 + k_{px} O_6 \quad (39)$$

4.2 PIDNNS2 controller

The architecture with at least one feedback link, as seen in Fig. 4. The feedback architecture, also known as a dynamic neural network, has one or more feedback links whose state changes over time in each neuron. The network's capacity for learning and

Table 3. Define the component of PIDNNS1

Comment	Description
I_i, H_j, O_k	Output of Layer 1, layer2, and layer3
W_{ji}	Weights between layer1, and layer2
V_{kj}	Weights between layer2, and layer3
b_j, b_k	Bias weight for layer2, and layer3
f_j	The activation function is the hyperbolic tangent function

performance is significantly impacted by the presence of a feedback connection. The state of a neuron in a feedback neural network depends not only on the current input signal but also on prior states of the neuron because feedback neural networks' weights are programmable [22].

Layer 1:

$$I_i(n) = [e_\theta(n), e_x(n)] \quad (40)$$

Layer 2:

$$net_1(n) = kp_\theta\big(e_\theta(n)\big) \quad (41)$$

$$net_2(n) = ki_\theta \cdot st \cdot e_\theta(n) + I_\theta(n-1) \quad (42)$$

$$net_3(n) = kd_\theta\big(e_\theta(n) - e_\theta(n-1)\big)/st \quad (43)$$

$$net_4(n) = kp_\theta\big(e_x(n)\big) \quad (44)$$

$$net_5(n) = ki_x \cdot st \cdot e_x(n) + I_x(n-1) \quad (45)$$

$$net_6(n) = kd_x\big(e_x(n) - e_x(n-1)\big)/st \quad (46)$$

Layer 3:

$$H_j(n) = f_j\big(\sum_i W_{ji} \cdot net_i(n)\big) + H_j(n-1) + h_j(\mathrm{n}-1) \quad (47)$$

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (48)$$

Layer 4:

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n)\right) + h_k(\mathrm{n}-1) \quad (49)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (50)$$

Layer 5:

$$O_l(n) = \sum_k Z_{lk}\, h_k(n) \quad (51)$$

Table 4. Define the component of PIDNNS2

Comment	Description
$I_i, net_i, H_j, h_k, O_l$	Output of Layer1, layer2, layer3,layer4,layer5
$kp_x, ki_x, kd_x, kp_\theta, ki_\theta, kd_\theta,$	Weights between layer1, and layer2
W_{ji}	Weights between layer2, and layer3
V_{kj}	Weights between layer3, and layer4
Z_{lk}	Weights between layer4, and layer5
f_j, g_k	Sigmoid activation function
st	Step size

Control signal:

$$u(n) = O_1(n) + O_2(n) \quad (52)$$

4.3 PIDNNS3 controller

The context nodes in the Elman network have self-connections, as in Fig. 5 which makes it also sensitive to the history of input data. This property is highly helpful in modeling dynamic systems. There have been other models put forth that enhance the original Elman networks. By introducing feedforward connections between the context nodes and the output nodes, the dynamic properties possess integral (I) dynamic features by nature, and convergence speed is improved [23].

Layer 1:

$$I_i(n) = [e_\theta(n), e_x(n)] \quad (53)$$

Layer 2:

$$net_1(n) = kp_\theta\big(e_\theta(n)\big) \quad (54)$$

$$net_2(n) = ki_\theta \cdot st \cdot e_\theta(n) + I_\theta(n-1) \quad (55)$$

$$net_3(n) = kd_\theta\big(e_\theta(n) - e_\theta(n-1)\big)/st \quad (56)$$

$$net_4(n) = kp_\theta\big(e_x(n)\big) \quad (57)$$

$$net_5(n) = ki_x \cdot st \cdot e_x(n) + I_x(n-1) \quad (58)$$

$$net_6(n) = kd_x\big(e_x(n) - e_x(n-1)\big)/st \quad (59)$$

Layer 3:

$$H_j(n) = f_j\left(\sum_i W_{ji} \cdot net_i(n)\right) + \sum_c Q_{jc}\, C_c^H(n) \quad (60)$$

Table 5. Define the component of PIDNNS3

Comment	Description
$I_i, net_i, H_j, h_k, O_l, C_c^H$	Output of Layer1, layer2, layer3,layer4,layer5, and Layer6
$kp_x, ki_x, kd_x, kp_\theta, ki_\theta, kd_\theta$	Weights between layer1, and layer2
W_{ji}	Weights between layer2, and layer3
V_{kj}	Weights between layer3, and layer4
Q_{jc}	Weights between context, and layer3
R_{kc}	Weights between context, and layer4
Z_{lk}	Weights between layer4, and layer5
f_j, g_k	Sigmoid activation function
α	Feedback gained from the self connections
st	Step size

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (61)$$

Layer 4:

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n)\right) + \sum_c R_{kc}\, C_c^H(n) \quad (62)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (63)$$

Layer 5:

$$O_l(n) = \sum_k Z_{lk}\, h_k(n) \quad (64)$$

Layer 6: context layer

$$C_c^H(n) = H_j(n-1) + \alpha C_c^H(n-1) \quad (65)$$

Control signal:

$$u(n) = O_1(n) + O_2(n) \quad (66)$$

4.4 PIDNNS4 controller

The controller Structure4 (S4) is shown in Fig. 6. The Jordan-Elman recurrent neural network (RNN), changes the fundamental RNN architecture by including a feedback connection from the network's output back into the hidden layer. This feedback connection speeds up learning by enabling the network to handle sequential data more skillfully and to keep a short-term recall of prior inputs [24].

Layer 1:

$$I_i(n) = [e_\theta(n), e_x(n)] \quad (67)$$

Layer 2:

$$net_1(n) = kp_\theta\big(e_\theta(n)\big) \quad (68)$$

$$net_2(n) = ki_\theta \cdot st \cdot e_\theta(n) + I_\theta(n-1) \quad (69)$$

$$net_3(n) = kd_\theta\big(e_\theta(n) - e_\theta(n-1)\big)/st \quad (70)$$

$$net_4(n) = kp_\theta\big(e_x(n)\big) \quad (71)$$

$$net_5(n) = ki_x \cdot st \cdot e_x(n) + I_x(n-1) \quad (72)$$

$$net_6(n) = kd_x\big(e_x(n) - e_x(n-1)\big)/st \quad (73)$$

Layer 3:

$$(n) = f_j\left(\sum_i W_{ji} \cdot net_i(n) + \sum_c Q_{jc}\, C_c(n) + \sum_m R_{jm} J_m(n)\right) \quad (74)$$

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (75)$$

Layer 4:

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n) + \sum_r E_{kr}\, S_r(n)\right) \quad (76)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (77)$$

Layer 5:

$$O_l(n) = \sum_k Z_{lk} \cdot h_k(n) \quad (78)$$

Layer 6:

$$C_c(n) = H_j(n-1) + \alpha C_c^H(n-1) \quad (79)$$

Layer 7:

$$J_m(n) = h_k(n-1) + \beta J_m(n-1) \quad (80)$$

Layer 8:

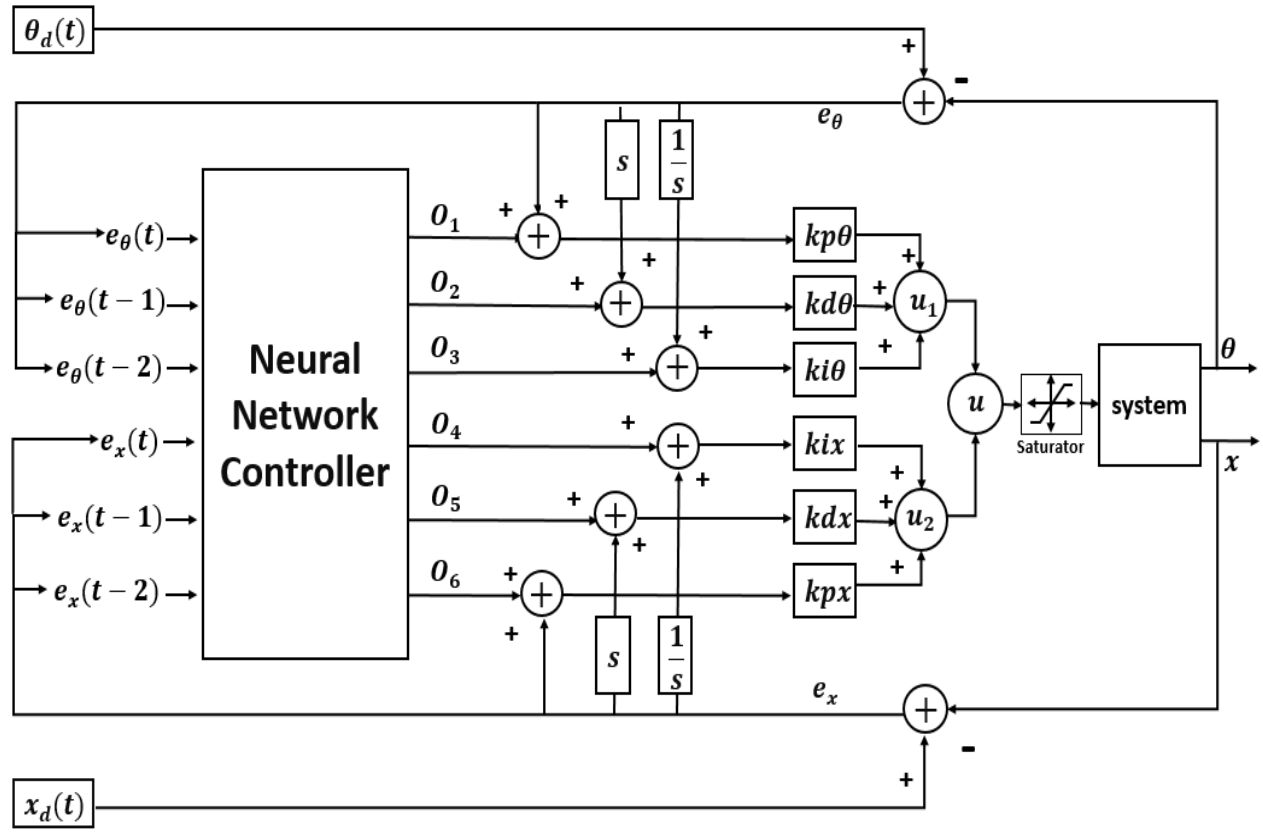
$$S_r(n) = h_k(n-1) + \eta S_r(n-1) \quad (81)$$

Control action:

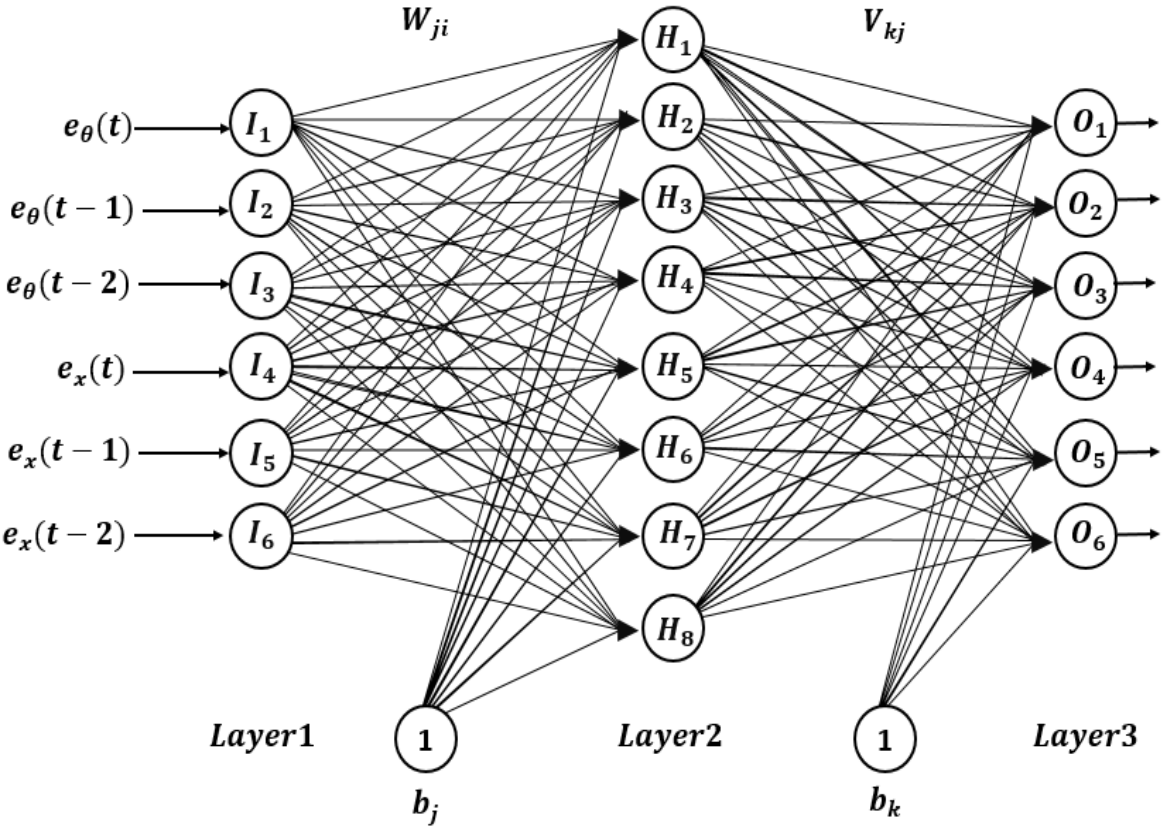
$$u(n) = O_1(n) + O_1(n) \quad (82)$$

5. Simulations results and performance analysis

The trajectory tracking for the IP system (nominal case), The suggested controllers are simulated using a MATLAB program with a step size of 0.001. The



(a)



(b)

Figure. 3 (a) PIDNNS1 controller, (b) Neural network controller

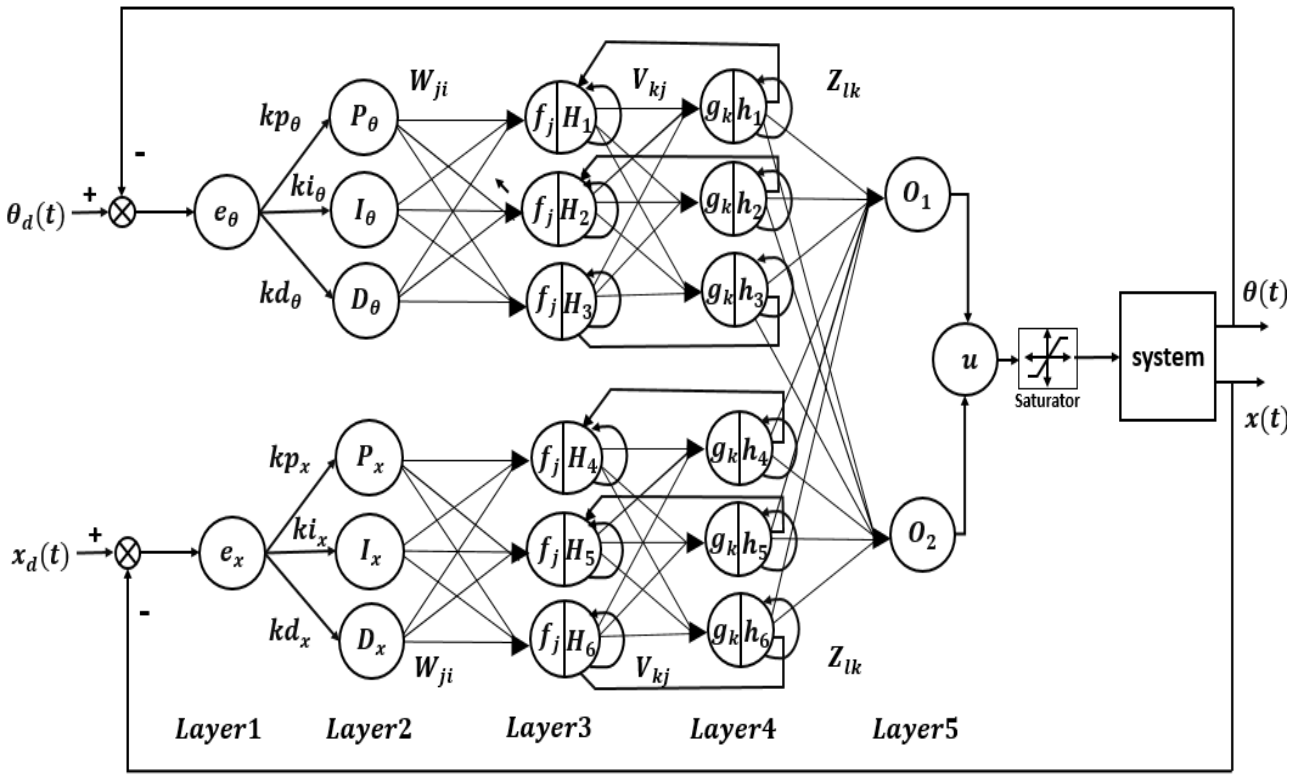


Figure. 4 PIDNNS2 controller

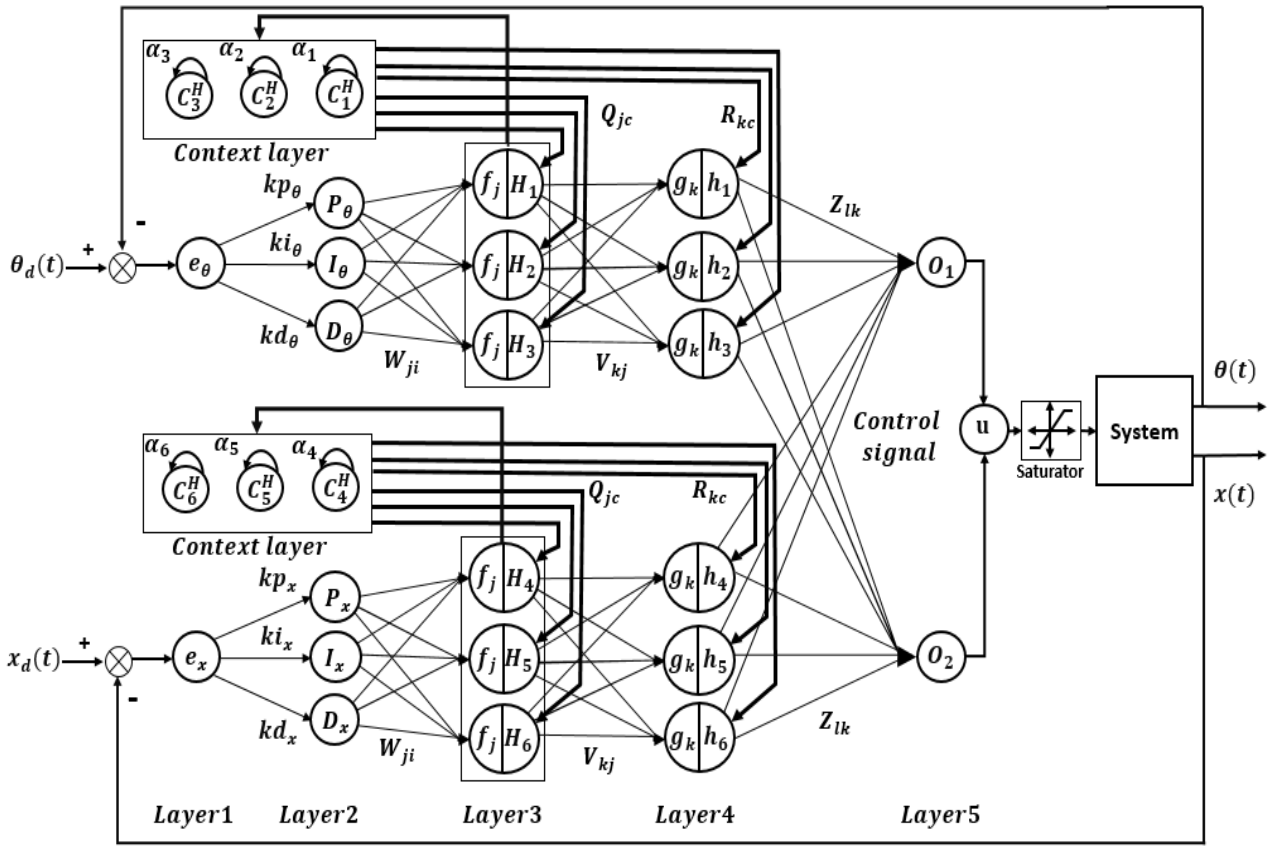
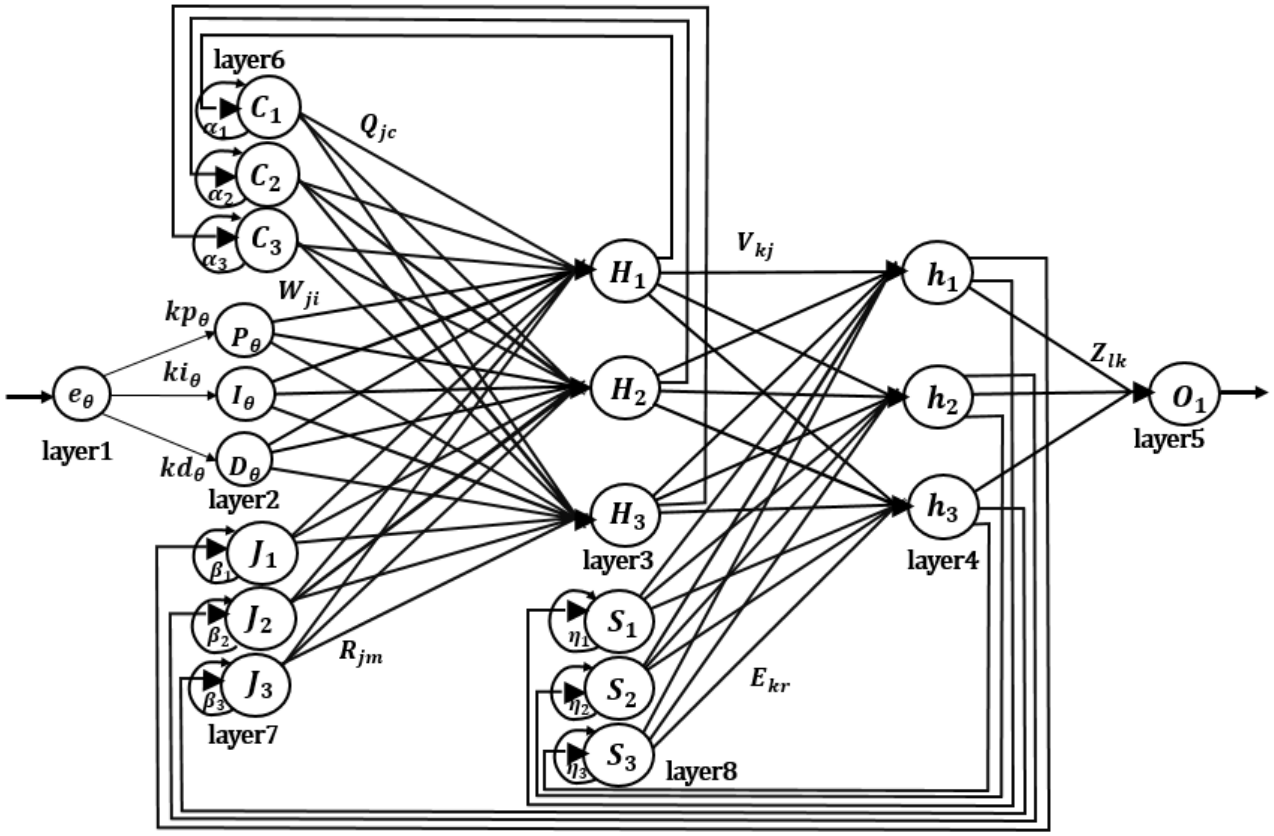
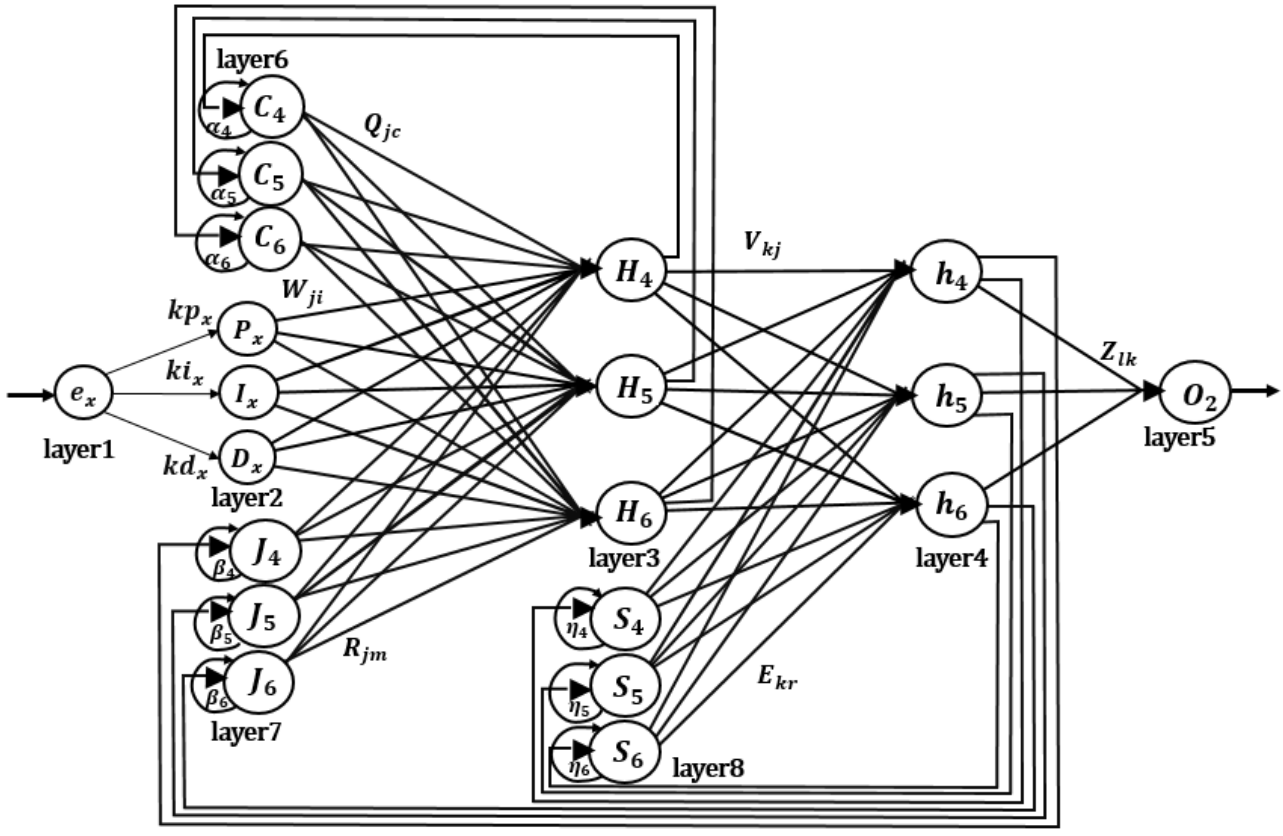


Figure. 5 PIDNNS3 controller

Received: September 5, 2023. Revised: September 28, 2023.



(a)



(b)

Figure.6 (a) PIDNNS4 Controller (a) PIDNNS4 Controller1, (b) PIDNNS4 Controller2

Table 6. Define the component of PIDNNS4

Comment	Description
$I_i, net_i, H_j, h_k, O_l, C_c J_m, S_r$	Layer 1, Layer2, Layer3, Layer4, Layer5, Layer6, Layer7, and Layer8
$kp_x, ki_x, kd_x, kp_\theta, ki_\theta, kd_\theta$	The weights between Layer1, and Layer2
W_{ji}	Weights between Layer2, and Layer3
V_{kj}	Weights between Layer3, and Layer4
Z_{lk}	Weights between Layer4, and Layer5
Q_{jc}	Weights between Layer3, and Layer6
R_{kc}	Weights between Layer3, and Layer7
E_{kr}	Weights between Layer4, and Layer8
α, β, η	Feedback gain to the self-connection of Layer6, Layer7, and layer8
f_j, g_k	Sigmoid activation function
st	Step size

Table 7. Weight of the cost function used with IP

weights	value
w_1	1
w_2	1
w_3	0.05

Table 8. Parameters of ACO used with IP

Parameters	Value
Population size (m)	40
Size of the archive (k).	100
Intensification factor (q)	0.5
Deviation-distance ratio (ξ)	1

Table 9. Parameters of controllers used with IP

Parameters	Value
Number of parameters	PIDNNS1=118
	PIDNNS2=56
	PIDNNS3=80
	PIDNNS4=128
kp, kd, ki	$[-150, 150]$
Weight of the neural network	$[-1, 1]$

proposed PIDNN controller's commands can be altered to satisfy design requirements and provide the user with a variety of control limit alternatives. Additionally, the cost function (J) performance index is used in the test.

$$J = \int_0^T [\, w_1 \times \theta^2(t) + w_2 \times x^2(t) + w_3 \times u^2(t)]dt \quad (83)$$

The weights of the cost function are illustrated in

Table 10. Cost function of PIDNN for tracking IP

Controller	Cost function (J)
PIDNNS1	1.189836
PIDNNS2	1.218626
PIDNNS3	1.185051
PIDNNS4	1.177494

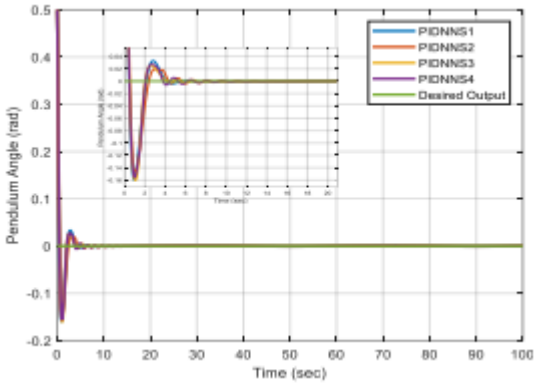


Figure. 7 Pendulum angle for trajectory tracking of IP

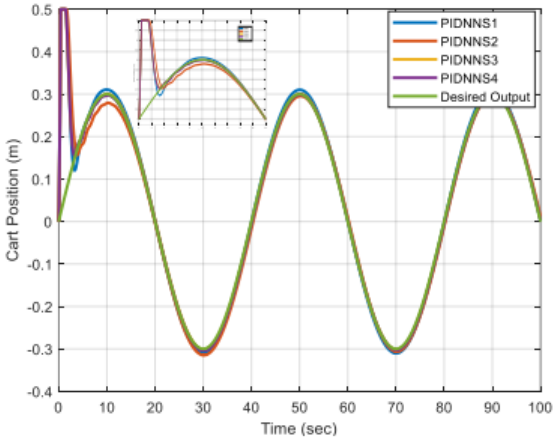


Figure. 8 Cart position for trajectory tracking of IP

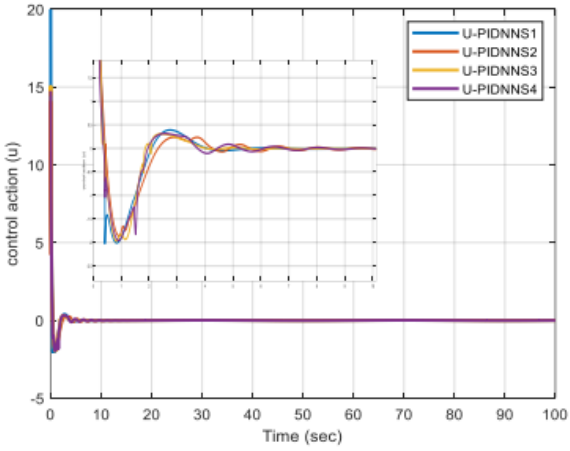


Figure. 9 Control action for trajectory tracking of IP

Table 11. Cost function of PIDNN for stabilization IP

Controller	Cost function (J)
PIDNNS1	1.280839
PIDNNS2	1.457746
PIDNNS3	1.301771
PIDNNS4	1.284467

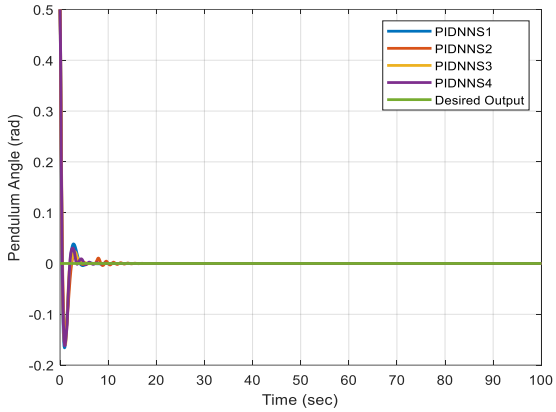


Figure. 10 Pendulum angle for stabilization of IP

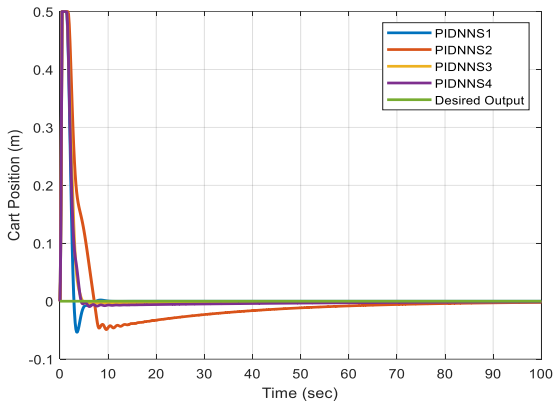


Figure. 11 Cart position for stabilization of IP

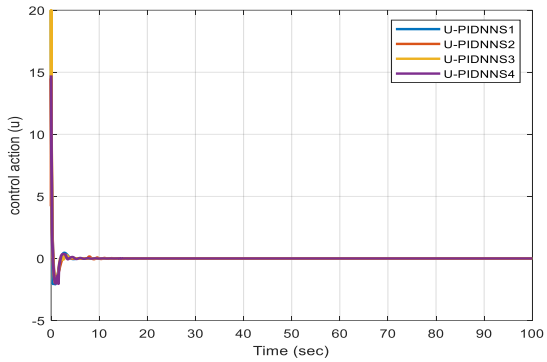


Figure. 12 Control action for stabilization of IP

Table 7. In terms of the starting point, the parameters of ACO optimization are illustrated in Table 8. All algorithms have the same parameters and the controller's parameter in Table 9.

Pendulum angle $\theta(t)$ and cart position $x(t)$ have ranges of $\{-0.5\,,0.5\}rad$ and $\{-0.5\,,0.5\}m$, and u(t) $\{-20\,,20\}N$ respectively. The IP system initial condition $\theta = 0.5\,rad\,, x = 0m$ Output desired $\theta_d = 0\,rad\,,\ x_d = 0.3\sin(0.05\pi t)\,m\,,$ and the simulation time is set to $100\,sec$. Figs. 7, 8, and 9 show the simulation results for controllers. The corresponding cost function (J) of these controllers as illustrated in Table 10.

In this case, (θ) is minimum settling time, overshoot in the event of PIDNNS3, and PIDNNS2, respectively. (x) is minimum settling time, overshoot, and oscillation in the PIDNNS3, the minimum cost function (J) in the PIDNNS4 controller.

5.1 Robustness analysis of the controllers

a. Stabilization

In this case, assuming initial conditions $= 0.5\,rad\,, x = 0m\,,$ the output desired $\theta_d = 0\,rad\,,$ $x_d = 0m,$ and simulation time is set to $100\,sec$. The inverted pendulum system's reaction under the PIDNN controllers is shown in Figs. 10, 11 and 12. The corresponding cost function (J) of these controllers is illustrated in Table 11.

In this test, (θ) is the minimum settling time, overshoot in the event of PIDNNS4, and PIDNNS2, respectively. (x) is the minimum settling time, overshoot, and oscillation in the PIDNNS3, the minimum cost function (J) in the PIDNNS1 controller.

b. Uncertainty in pendulum mass

Although the pendulum mass is uncertain, this test is similar to the previous one (nominal case) but with uncertainty in pendulum mass. One potential source of uncertainty is a 50% increase in pendulum mass. Figs. 13, 14, and 15 show pendulum performance with PIDNN controllers in the presence of this uncertainty. The corresponding cost function (J) of these controllers is illustrated in Table 12.

In this test, (θ) is minimum settling time, overshoot in the event of PIDNNS3, and PIDNNS2, respectively. (x) is the minimum settling time, overshoot, and oscillation in the PIDNNS3, the minimum cost function (J) in the PIDNNS4 controller.

c. External disturbance

Following the addition of the angle disturbance value equal to 0.06 rad at a time equal to 30 sec, the response of the inverted pendulum system using the PIDNN controllers is depicted in Figs. 16, 17, and 18. the corresponding cost function (J) of these controllers as in Table 13.

Table 12. Cost function of PIDNN for the uncertainty IP

Controller	Cost function (J)
PIDNNS1	1.290245
PIDNNS2	1.606191
PIDNNS3	1.281229
PIDNNS4	1.273627

Table 13. Cost function of PIDNN for disturbance of IP

Controller	Cost function (J)
PIDNNS1	1.242601
PIDNNS2	26.358405
PIDNNS3	1.222391
PIDNNS4	1.209761

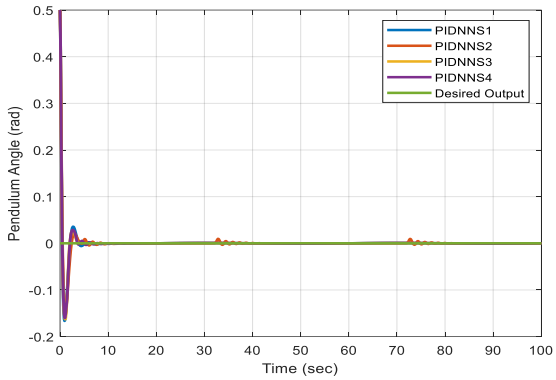


Figure. 13 Pendulum angle for the uncertainty of IP

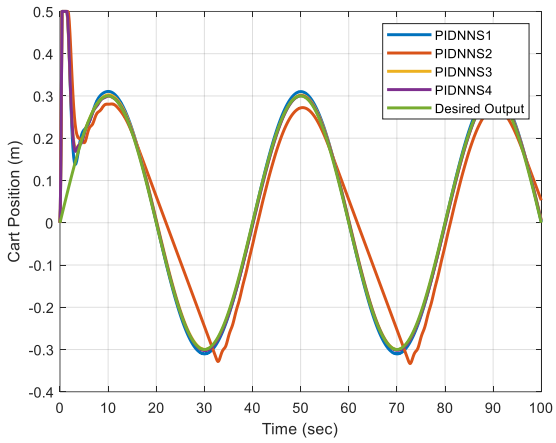


Figure. 14 Cart position for the uncertainty of IP

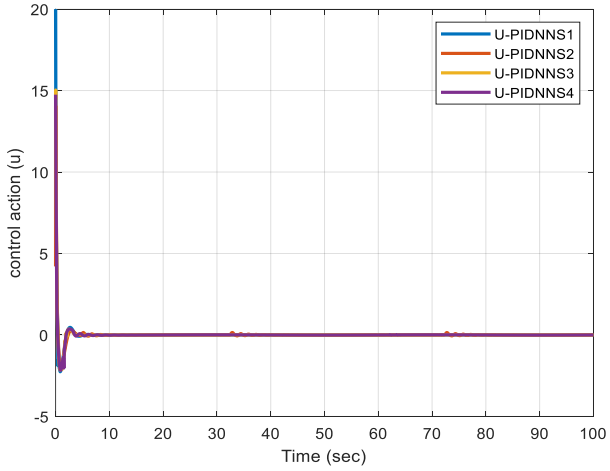


Figure. 15 Control action for the uncertainty of IP

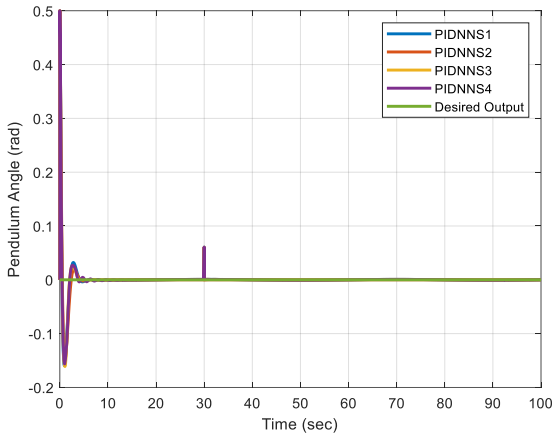


Figure. 16 Pendulum angle for disturbance of IP

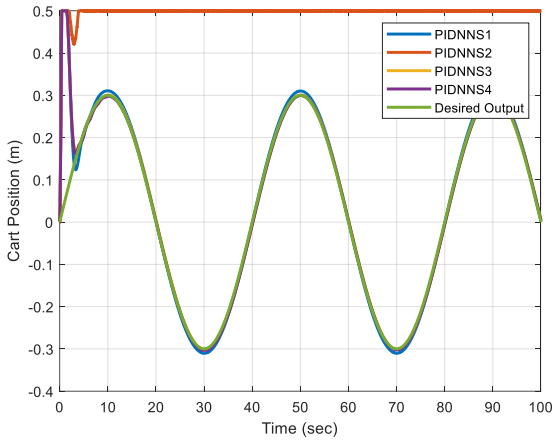


Figure. 17 Cart position for disturbance of IP

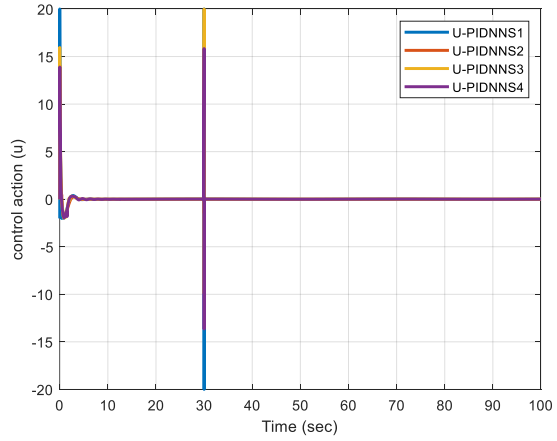


Figure. 18 Control action for disturbance of IP

Table 14. Comparing the values of the cost function

Controller	Cost function
PIDNNS1 in a case study.	1.189836
PIDNNS2 in a case study.	1.218626
PIDNNS3 in a case study.	1.185051
PIDNNS4 in a case study.	1.177494
PID - PSO in [9].	1.8449
FOPID-PSO in [9].	1.7131

In this test, (θ) is minimum settling time, overshoot in the event of PIDNNS3. (x) is minimum settling time, overshoot, and oscillation in the PIDNNS3, and PIDNNS2 is unstable. the minimum cost function (J) in the PIDNNS4 controller.

5.2 Discussion and comparison

Current study outperforms other controllers such as PID controller with optimization for particle swarms (PID-PSO), FOPID controller with optimization for particle swarm (FOPID-PSO) in terms of trajectory tracking, lower cost function, and energy efficiency , this aspect is crucial. The results indicate that proposed methodology achieves superior performance compared to these controllers. Additionally, current study shows that the system reaches stability faster and experiences reduced overshoot compared to the other controllers. This characteristic is essential for ensuring stable and precise control of the inverted pendulum system. This work's primary contribution is to lower the cost function and energy. Table 14 shows the corresponding cost function (J) of trajectory tracking for various controllers compare with PID, and FOPID controllers.

6. Conclusion

In this work, four PIDNN controller structures were proposed for controlling an inverted pendulum (IP) system. The metaheuristic ACOR is used to adjust the controller parameters and neural network weights. In terms of model uncertainty, disturbance rejection, and initial circumstances, the robustness of these controllers has also been examined. The results show that the PIDNN controllers, of which the best is the PIDNNC4, have a respectable ability to quickly minimize the variation between real and desired routes without chattering in control signals.

Conflicts of interest

The authors declare that they have no conflicts of interest. This research was conducted independently and without funding from any external sources.

Author contributions

Conceptualization, Zaid S. Mohsen; methodology, Zaid S. Mohsen ; software, Mohamed Jasim Mohamed; validation, Zaid S. Mohsen; formal analysis, Zaid S. Mohsen; investigation, Zaid S. Mohsen; resources, Zaid S. Mohsen ; data curation, Zaid S. Mohsen; writing—original draft preparation, Zaid S. Mohsen; writing—review and editing, Zaid S. Mohsen; visualization, Zaid S. Mohsen; supervision, Mohamed Jasim Mohamed; project administration, Mohamed Jasim Mohamed; funding acquisition, Not funded.

Acknowledgments

The authors acknowledge The Department of Control and Systems Engineering at the University of Technology in Iraq.

List of notations

Abbreviation	Description
IP system	
m	Mass of pendulum
M	Mass of cart
l	Length of a bar
g	Standard gravity
$x_1 = \theta$	the angle of the pendulum
$x_2 = \dot{\theta}$	the angular velocity
$x_3 = x$	is the location of the cart
$x_4 = \dot{x}$	is the velocity of the cart
$u = F$	Control action.
d_1, d_2	disturbance
Ant Colony Optimization	
$G^i(x)$	Gaussian functions
σ_l^i	standard deviation
μ_i^i	the mean
P_l	probability
m	Population size
k	Size of the archive
q	Intensification factor
ξ	Deviation-distance ratio
Controller	
Kp	Proportional Gain
Kd	Derivative Gain
Ki	Integral Gain
Cost function	
J	the cost function performance index

References

[1] I. Shafi, Z. Malik, S. Din, G. Jeon, and J. Ahmad, "A computationally intelligent neural network-based nonlinear autoregressive exogenous balancing approach for real-time processing in industrial applications using big data", *Concurr. Comput. Pract. Exp.*, Vol. 33, No. 22, 2021.

[2] S. Alimoradpour, "Provide A Method Based on Genetic Algorithm to Optimize the Fuzzy Logic Controller for the Inverted Pendulum", *Research Square*, pp. 0–19, 2021.

[3] A. A. Santiago and G. B. D. A. Lima, "Teaching Intelligent Control: the Inverted Pendulum Experiment Revisited", *Rev. Ensino Eng.*, Vol. 39, No. 1, pp. 89–98, 2020.

[4] T. Johnson, S. Zhou, W. Cheah, W. Mansell, R. Young, and S. Watson, "Implementation of a Perceptual Controller for an Inverted Pendulum Robot", *J. Intell. Robot. Syst. Theory Appl.*, Vol. 99, No. 3–4, pp. 683–692, 2020.

[5] A. Mourad, Y. Zennir, and C. Tolba, "Intelligent and Robust Controller Tuned with WOA: Applied for the Inverted Pendulum", *J. Eur. des Syst. Autom.*, Vol. 55, No. 3, pp. 359–366, 2022.

[6] N. K. Nguyen, V. N. Pham, T. C. Ho, and T. M. P. Dao, "Designing an Effective Hybrid Control Strategy to Balance a Practical Inverted Pendulum System", *Int. J. Eng. Trends Technol.*, Vol. 70, No. 5, pp. 80–87, 2022.

[7] Q. Sun, C. Du, Y. Duan, H. Ren, and H. Li, "Design and application of adaptive PID controller based on asynchronous advantage actor–critic learning method", *Wirel. Networks*, Vol. 27, No. 5, pp. 3537–3547, 2021.

[8] A. Jain, A. Sharma, V. Jately, B. Azzopardi, and S. Choudhury, "Real-Time Swing-Up Control of Non-Linear Inverted Pendulum Using Lyapunov Based Optimized Fuzzy Logic Control", *IEEE Access*, Vol. 9, pp. 50715–50726, 2021, doi: 10.1109/ACCESS.2021.3058645.

[9] S. K. Mishra and D. Chandra, "Stabilization and Tracking Control of Inverted Pendulum Using Fractional Order PID Controllers", *Journal of Engineering*, Vol. 2014, 2014, doi: 10.1155/2014/752918.

[10] K. E. Dagher, "On-Line Tuning Sliding Mode Controller Design for Nonlinear Inverted Pendulum System based on Bees Algorithm", *Engineering and Technology Journal*, Vol. 34, No. 8, 2016, doi: 10.30684/etj.34.8A.9.

[11] P. Gautam, "System identification of nonlinear Inverted Pendulum using artificial neural network", In: *Proc. of 2016 Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2016*, pp. 0–4, 2016.

[12] H. I. Ali and M. J. Kadhim, "H 2 Sliding Mode Controller Design for Mobile Inverted Pendulum System", *Iraqi Journal of Computers, Communications, Control & Systems Engineering*, Vol. 18, No. 2, pp. 17–29, 2018.

[13] P. Supervisors, Q. M. Zhu, and P. N. Faculty, "Development of U-model Enhanced Nonlinear Systems", *uwe-repository.worktribe*, 2018.

[14] A. Ingram and S. D. Publisher, "Visual Feedback Stabilisation of a Cart Inverted Pendulum A Item Type Thesis", *bradscholars.brad.ac.uk*, 2021.

[15] S. Fidanova, *Studies in Computational Intelligence 947 Ant Colony Optimization and Applications*, 2020.

[16] T. K. Nguyen, I. G. Lee, O. Kwon, Y. J. Kim, and I. P. Hong, "Metaheuristic optimization techniques for an electromagnetic multilayer radome design", *J. Electromagn. Eng. Sci.*, Vol. 19, No. 1, pp. 31–36, 2019.

[17] J. Tavoosi, "A new general type-2 fuzzy predictive scheme for PID tuning", *Appl. Sci.*, Vol. 11, No. 21, pp. 1–15, 2021.

[18] A. J. Humaidi and A. S. M. A. Obaidi, "PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot", *International Review of Applied Sciences and Engineering*, 2023.

[19] M. Mohamed and M. Hamza, "Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO", *Eng. Technol. J.*, Vol. 37, No. 12A, pp. 574–583, 2019.

[20] O. F. Lutfy, "A Wavelet Functional Link Neural Network Controller Trained by a Modified Sine Cosine Algorithm Using the Feedback Error Learning Strategy", *Journal of Engineering Science and Technology*, Vol. 15, No. 1, pp. 709–727, 2020.

[21] A. Mohammadi and J. C. Ryu, "Neural network-based PID compensation for nonlinear systems: ball-on-plate example", *Int. J. Dyn. Control*, Vol. 8, No. 1, pp. 178–188, 2020.

[22] A. Y. A. Bakri and M. Sazid, "A Modified Elman Neural Network Model with Application to Dynamical Systems Identification", *Mining*, Vol. 1, No. 3, pp. 315–334, 2021.

[23] Y. An, X. Sun, B. Ren, H. Li, and M. Zhang, "A data-driven method for IGBT open-circuit fault diagnosis for the modular multilevel converter based on a modified Elman neural network", *Energy Reports*, Vol. 8, pp. 80–88, 2022, doi: 10.1016/j.egyr.2022.08.024.

[24] G. D. Sen, G. O. Gunel, and M. Guzelkaya, "Extended Kalman Filter Based Modified

Elman-Jordan Neural Network for Control and Identification of Nonlinear Systems", In: *Proc. of 2020 Innov. Intell. Syst. Appl. Conf. ASYU 2020*, No. 1, 2020.