Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 10, No. 4, December 2022, pp. 983~995
ISSN: 2089-3272, DOI: 10.52549/ijeei.v10i4.4216

ANFIS multi-tasking algorithm implementation scheme for ball-on-plate system stabilization

Oussama Hadoune[1], Mohamed Benouaret[2]
[1,2]Department of Electronics, LASA Laboratory, Faculty of Technology, University of Badji Mokhtar Annaba, Algeria

Article Info

Article history:

Received Oct 10, 2022
Revised Dec 22, 2022
Accepted Jan 2, 2023

Keywords:

Parallel robot
ANFIS
Multitasking
Ball-on-plate
Neural-network

ABSTRACT

This paper presents the design and realization of a ball-on-plate system using a 3-degree-of-freedom parallel robot controlled by an adaptive neuro-fuzzy inference system. The ball-on-plate system is nonlinear, multivariable, with an under-actuated feature. Initially, the parallel robot is designed using SolidWorks and mechanized using a computer numerical control machine. Followed by the presentation of the ball-on-plate system mathematical model and the simplified model obtained. Afterwards, the inverse kinematics are performed to derive the appropriate angle for each servomotor. Eventually, the controller is designed and implemented in a double loop feedback scheme. A comparison between the proposed controller and a conventional proportional–integral–derivative controller in terms of time response, overshoot, and steady-state error is carried out. Furthermore, a comparison between sequential and asynchronous parallel processing is conducted for two different scenarios. The first scenario is when moving the ball to the origin while the second is for disturbance rejection. Simulation and experimental results show that the adaptive neuro-fuzzy inference system implemented using asynchronous parallel processing improves the real-time system stability by considerably decreasing oscillations as well as enhancing the ball movement smoothness with a small stead-state error.

Corresponding Author:

Oussama HADOUNE
Department of Electronics, Faculty of Technology
University of Badji Mokhtar Annaba
Bp 12, 23000 Annaba, Algeria
Email: oussama.hadoune@univ-annaba.org

1. INTRODUCTION

Parallel robots are often used in different areas such as industrial practices, flight simulators, and medical science due to their high stiffness (the ability to lift a heavy weight that is shared by multiple legs), high-speed task execution (due to low moment of inertia), and their ability to solve high-precision problems. The ball-on-plate system (BPS) is generally used on 2-Degrees of Freedom (DOF), which is used as an educational platform where controllers can be tested. This type of ball-on-plate system is mainly used by schools and research institutes to help improve students' learning process. In this regard, the ball-on-plate system can be an excellent educational kit to understand nonlinear control, uncertainties, and controller design. A rotary pendulum system is a benchmark problem as well; however, it can be dangerous if we lose control over it. Therefore, the ball-on-plate system is the most suitable platform on which controllers can be tested safely.

Several research papers on the ball-on-plate system have already been published. Tudic et al. [1] conducted a comparative study between PID and a PD controllers in which they were applied to a 2-DOF ball balancer that was designed and manufactured using a 3D printer. Note that both controllers were implemented on an Arduino platform. The experimental results demonstrate that the PD algorithm prove to be more successful than the PID controller. Furthermore, Chevalier et al. [2] used a ball-on plate system on a 6-DOF Stewart platform controlled by an automatic calibration with a robust PD controller to compensate for the characterized dynamic model changes by the use of balls with different surfaces, masses, and diameters without

changing the controller's parameters. The results show the ability of the proposed approach to maintain the ball in the desired position despite the significant variations of the mathematical model. Similarly Rossel et al. [3] concentrated on the problems encountered in the real-time vision-based control as well as the assembly imperfection of the built 6-DOF Stewart platform that was controlled by a PD controller. The findings demonstrate that the proposed controller has a good tracking performance. Yaovaja [4] used a Fuzzy logic controller (FLC) to supervise the PID parameters applied to a ball on plate system on a 6-DOF Stewart platform. The results prove that the proposed combined controllers improve the time response. Mohammadi and Ryu [5] presented a neural network (NN)-based-PID that uses a backpropagation learning algorithm to control a ball on plate system. The controller's output is the sum of PID and NN-based-PID controllers. The results confirm that the proposed approach has a good dynamic and static tracking.

Kassam, Haddad and Albitar [6] proposed four control strategies, PID, Sliding mode control (SMC), FLC, and Linear–quadratic-regulator (LQR). The results demonstrated that SMC could overcome the changes in model dynamics and uncertainties. Negash and Singh [7] presented a simulation of a fuzzy logic controller combined with SMC to reduce the chattering effect. The simulation results show a good tracking performance with insignificant chattering. In addition, Ma, Tao and Huang [8] presented an observer integrated back-stepping control for the cascaded ball and plate system. Simulation and experimental results prove the ability of the proposed technique to overcome uncertainties. He and Cheng [9] utilized a loop transfer function recovery (LTR) and a nonlinear LTR control strategies to solve the problem of large overshoot and long adjustment time encountered when using a PID controller. The simulation results show the effectiveness of the presented strategy. Mamdani's maximum product FLC was designed to prove its robustness against nonlinearities and uncertainties. The results demonstrate the quality of the controller's performance in static and dynamic tracking [10]. Okafor et al. [11] introduced a comparative analysis between PID, Genetic algorithm (GA) PID, reinforcement learning (RL)-PID, deep deterministic policy gradient (DDPG) and two variants of deep reinforcement learning-based PID (DDPG-FC-350-R-PID and DDPG-FC-350-E-PID) to test the system's stability where DDPG-FC-350-E-PID created by this author, demonstrate the best performance in tracking control. Finally, Betancourt, Alarcon and Velasquez [12] presented a comparative study between a fuzzy control algorithm and a PID controller. The results demonstrate the FLC's ability to adapt to changes in the system's dynamics.

In literature, most of the presented ball-on-plate systems have 6- or 2-DOF. To the best of our knowledge, this is the first time that a 3-DOF parallel robot that uses three spherical joints has been designed to test the performance of the adaptive neural network-based fuzzy inference system (ANFIS), using a hybrid learning algorithm against uncertainties, nonlinearities, and for enhancing the system's response compared to that of a conventional PID controller. The choice of fuzzy rules and membership functions is time-consuming and requires an in-depth knowledge and experience in working with the fuzzy control design. On the other hand, ANFIS solves the problem of defining rules and membership functions with a high adaptation capability and rapid learning ability [13]. Further, to enhance the system's response and due to the complexity of the real-time algorithm implementation, we used a dual-core microcontroller on which the developed tasks are split into two cores to reduce the algorithmic complexity. The latter has demonstrated a real improvement in terms of oscillations and system's stability. Additionally, a disturbance rejection test has been proposed to evaluate the effectiveness of the proposed techniques. Therefore, the main contribution of this paper is to demonstrate the effect of parallel and sequential processing on conventional and adaptive controllers.

The rest of this manuscript is organized as follows. Section 2 presents the off-line processing. In section 3, the inverse kinematics and the mechanical design are demonstrated. Section 4 details the proposed topology and the controller's design. In Section 5, the simulation and experimental validation of the real-time control are presented, followed by the concluding remarks in section 6.

2. SYSTEM MODELLING AND IDENTIFICATION

The structure of the system is made of two parts as follows: a fixed and a moving platform, connected by several serial chains called legs. The fixed part carries the servomotors linked to the mobile part via three axes. The ball rolls in X and Y directions. Based on [14], the plant parameters description is presented in Table 1. and the mathematical model of the BPS is obtained as follows:

$$L = T - V \tag{1}$$

The kinetic energy T is the sum of the ball and the plate's kinetic energies and is given by:

$$T = T_b + T_p \tag{2}$$

Where:

$$T_b = \frac{1}{2}(m_b + \frac{I_b}{r^2})(\dot{x}_b^2 + \dot{y}_b^2) \quad (3)$$

Since the ball is rolling without slippage, its linear velocities are described as follows:

$$\dot{x}_b^2 = r_b + w_y \quad , \quad \dot{y}_b^2 = r_b + w_x \quad (4)$$

Substituting the linear velocities equations in the kinetic energy equation, we obtain:

$$T_b = \frac{1}{2}(m_b + \frac{I_b}{r^2})(\dot{x}_b^2 + \dot{y}_b^2) \quad (5)$$

The plate kinetic energy expression is described by:

$$T_p = \frac{1}{2}\left(I_p + I_b\right)\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b(x_b\dot{\alpha} + y_b\dot{\beta})^2 \quad (6)$$

The potential energy equation of the plate is given by:

$$V = m_b g(x_b \sin\alpha + y_b \sin\beta) \quad (7)$$

Replacing equation 1 into the following Euler equation:

$$\frac{d}{dt}\frac{\partial T}{\partial qi} - \frac{d}{dt}\frac{\partial T}{\partial qi} + \frac{d}{dt}\frac{\partial V}{\partial qi} = \partial Qi \quad (8)$$

We obtain:

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b - m_b\left(x_b\dot{\alpha}^2 + y_b\dot{\alpha}\dot{\beta}\right) + m_b g \sin\alpha = 0 \quad (9)$$

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b - m_b\left(y_b\dot{\beta}^2 + x_b\dot{\alpha}\dot{\beta}\right) + m_b g \sin\beta = 0 \quad (10)$$

$$tx = \left(I_p + I_b + m_b x_b^2\right)\ddot{\alpha} + 2m_b x_b\dot{x}_b\dot{\alpha} + m_b x_b y_b\ddot{\beta} + m_b\dot{x}_b y_b\dot{\beta} + m_b x_b\dot{y}_b\dot{\beta} + m_b g x_b \cos\alpha \quad (11)$$

$$ty = \left(I_p + I_b + m_b y_b^2\right)\ddot{\beta} + 2m_b y_b\dot{y}_b\dot{\beta} + m_b x_b y_b\ddot{\alpha} + m_b\dot{x}_b y_b\dot{\alpha} + m_b x_b\dot{y}_b\dot{\alpha} + m_b g y_b \cos\beta \quad (12)$$

Designing a controller for such a complex nonlinear system is almost impossible. Therefore, as we mentioned earlier, the ball on plate system is simplified by ignoring equations 11 and 12 that represent the servomotor torque effect on the ball on plate system. Equations 9 and 10 are simplified. Since the plate inclination is small, we assume that: sin α=α, sin β=β, as well as the angular velocities: α=0 and β=0.
The obtained equations are:

$$\left(\frac{7}{5}\right)\ddot{x}_b + g\alpha = 0 \quad (13)$$

$$\left(\frac{7}{5}\right)\ddot{y}_b + g\beta = 0 \quad (14)$$

Both obtained equations have the plate inclination value as input and the ball position on the plate as output. Thus, as long as the transfer function is the same for both the X and Y axes, in the following sections, only the X-axis will be studied and we will apply the same to the Y-axis.

The simplified BPS model is not accurate enough. Hence, it will be used to approach the PID controller's parameters which will be slightly and manually tuned to determine their exact values.

Table 1. Plant parameters description

Parameters	Description
$(\dot{x}_b^2, \dot{y}_b^2)$	Ball's linear velocities along x and y axes
r_b	Radius of the ball
(w_x, w_y)	Ball's angular velocities alon x and y axes
m_b	Mass of the ball
I_b	Moment of inertia of the ball
I_p	Moment of inertia of the plate
$(\dot{\alpha}, \dot{\beta})$	Plate's angular velocities
(α, β)	Plate's angle of inclination along x-axis and y-axis
g	Gravitational acceleration

2.1. Servomotor model identification

The servomotor model is identified using experimental data through the MATLAB system identification toolbox, where the input of the transfer function is a pulse with modulation (PWM) and the output is the position angle. Different values of the input/output were recorded and introduced to the MATLAB toolbox. The obtained model is given by:

$$\frac{1.91}{0.0022s^2 + 0.16001s + 3.6561} \tag{15}$$

3. INVERSE KINEMATICS & MECHANICAL DESIGN

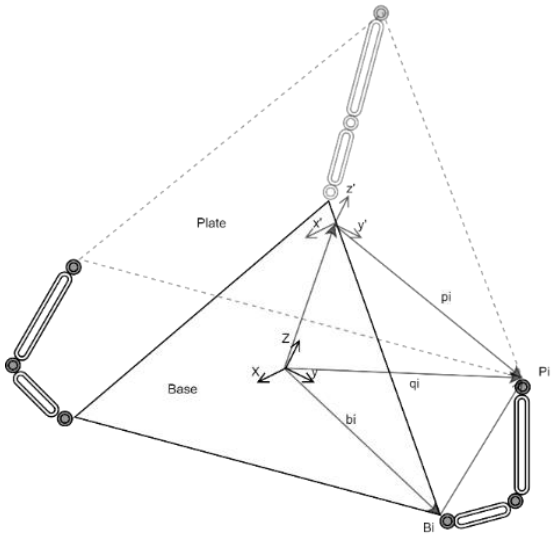

Figure 1. Stewart platform model

The built Stewart platform has a 3-DOF. Based on the X and Y coordinates (roll and pitch values), we can calculate the appropriate angle for each servomotor using inverse kinematics. According to Fig.1 the servomotors are placed in the middle of an equilateral triangle, separated by 120 degrees, We calculate the servomotor attachments position, where the base and plate attachments coordinates are defined as follows:

$$b_i = \begin{bmatrix} r_b & \frac{-1}{2}r_b & \frac{-1}{2}r_b \\ 0 & \sqrt{\frac{3}{2}r_b} & -\sqrt{\frac{3}{2}r_b} \\ 0 & 0 & 0 \end{bmatrix},\ p_i = \begin{bmatrix} r_b & \frac{-1}{2}r_p & \frac{-1}{2}r_p \\ 0 & \sqrt{\frac{3}{2}r_p} & -\sqrt{\frac{3}{2}r_p} \\ 0 & 0 & 0 \end{bmatrix}$$

Where the translational vector is defined as follows:

$$R = R_x(\theta)R_y(\emptyset)R_z(\varphi) \tag{16}$$

Where:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} R_y = \begin{bmatrix} \cos(\emptyset) & 0 & \sin(\emptyset) \\ 0 & 1 & 0 \\ -\sin(\emptyset) & 0 & \cos(\emptyset) \end{bmatrix} R_x = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

q_i is the location vector of the connections pi with respect to the base frame:

$$q_i = T + R_b^p \times p_i \tag{17}$$

Where the translational vector is defined as follows:

$$T = \begin{bmatrix} x \\ y \\ z + h_0 \end{bmatrix}$$

$h_0$represents the height of the platform relative to the base "home".
We compute the necessary leg lengths for linear actuators as following:

$$L_i = \sqrt{(q_{0i} - b_{0i})^2 + (q_{1i} - b_{1i})^2 + (q_{2i} - b_{2i})^2} \tag{18}$$

In our case, we use rotational actuators. Thus, the servomotor angle equation is:

$$\theta = \cos^{-1}\left(\frac{a^2 + s^2 + L_i^2}{L_i^2 + a^2}\right) \tag{19}$$

Where:
a denotes is the length of the servomotor arm.
s indicates the length of the arm connecting the servomotor arm to the platform.

3.1. Mechanical 3D design

SolidWorks-2020 is a computer-aided design (CAD) software that was used to design the Stewart platform. Instead of using prismatic actuators, we use rotary actuators which are more cost-effective. Fig.2. demonstrates the 3D designed model which is composed of a fixed base connected to the plate through servomotor horns as well as parallel arms (pushrods) that use spherical joints. Servomotors are placed on the base and are responsible for the movement of the plate. The various designed parts presented in Fig.3. were presented as vectors and their Gcode files were extracted and sent to a numerical control machine (CNC).

Figure 2. Stewart platform CAD model

Figure 3. CAD model for different parts of the stewart platform

3.2. Parallel robot setup

To make the model more robust and to limit the mechanical inaccuracies, servomotor pushrods are made of carbon fiber while the rest of the parts are made of alucobond (which is vibration-damping), except for the upper and the lower attachments which were made of aluminium. The lengths of the servomotor's rod and horn are 200mm and 60mm respectfully. To limit the plate movement vibrations, every two servomotors rods are placed in such a way that the plate movement stability increases. The final result is shown in Fig.4.

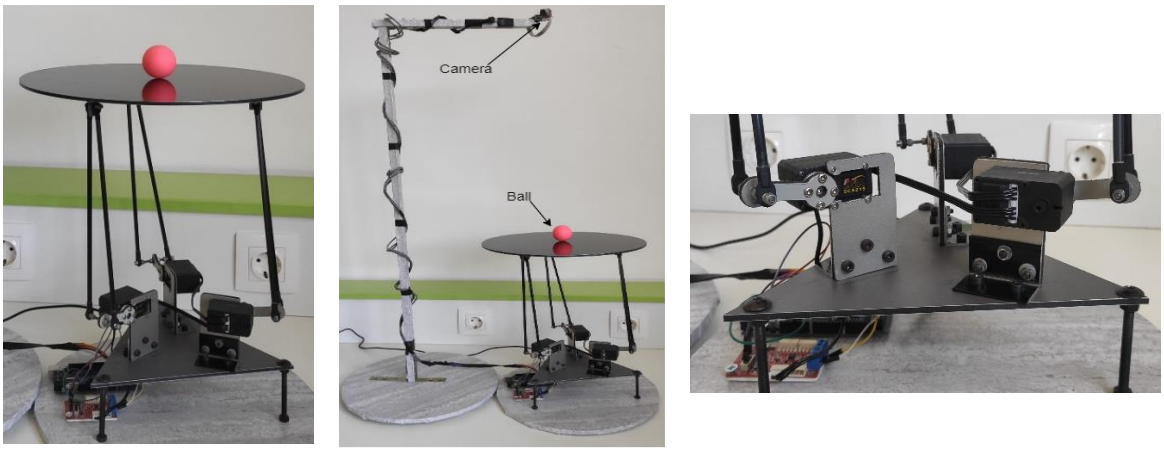

Figure 4. Experimental setup of the stewart platform

3.3. Hardware Setup

3.3.1. Servomotor

The Servomotor Feetech SCS215 is a serial bus smart servomotor that is used to balance the plate. This type of servomotor provides information on the position, current, voltage, velocity, and temperature with a resolution of 0.19 degrees. It is built with metal gears and integrates an asynchronous rs485 half-duplex serial communication protocol that can connect up to 253 Servomotor IDs at a speed of 1Mbps and a maximum torque of 17 kg.cm. The servomotor is controlled using an integrated P controller.

3.3.2. Camera

The Pixy2 is a small, fast, low-cost camera with powerful image processing capabilities. It can detect up to seven colours and indicate their coordinates in a 320 x 200-pixel graphic (x, y) at a speed of 60fps, which is fast enough to track a ball rolling on a plate [15], [16]. The main advantage of using a camera which takes care of image processing is the reduced consumption of resources. In our case, the chosen microcontroller only has the task of getting the ball's coordinates, which are then processed at the pixy level and introduced to the implemented controller. Thus, the resources consumed are minimum. Another advantage of using a Pixy camera over the touch panels usually used in ball on plate systems is the shape of the plate. Touch panels are frequently square or rectangular, which dictates the shape of the plate to be used. The use of a camera allows the use of an end effector of any shape.

3.3.3. Microcontroller

A 32-bit, low-cost microcontroller board based on the Tensilica Xtensa LX6 microprocessor in both dual core and single-core variations with a 240 MHz clock is used to test the proposed controllers [17]. It connects to the Pixy camera through the Serial Peripheral Interface (SPI) and to the servomotors via the Universal Asynchronous Reception and Transmission (UART) communication protocol.

3. ANFIS TOPOLOGY & CONTROLLER DESIGN

The double-loop feedback scheme is designed in a way that the inner loop aims to control the servomotor angle position, while the outer loop takes the input from the inner loop feedback to control the ball's position, as shown in Fig.5.

From [18], we can conclude that the BPS in an open-loop is a non-damped second-order system. In theory, this system can be classified as "marginally" stable, but in practice, is in fact unstable. Furthermore, to evaluate the BPS in a closed-loop, a Routh stability test is conducted. The author confirmed the system stability when cascading the proposed controller.

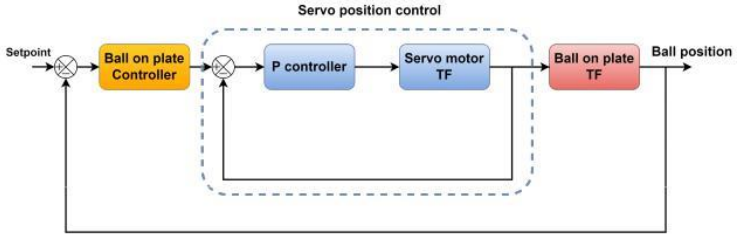

Figure 5. Double loop feedback scheme

Fig.6. illustrates the general control scheme of the designed parallel robot, which contains two controllers. The first one stabilizes the ball along the X-axis, while the second one stabilizes the ball along the Y-axis. The controller's outputs are fed to the inverse kinematics algorithm, which allows for control over three servomotors.

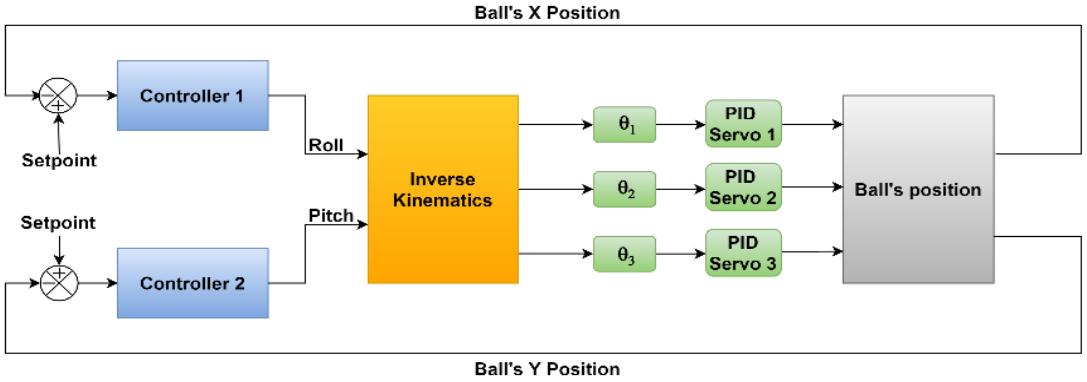

Figure 6. The control scheme of the designed parallel robot

4.1. Adaptive neuro-fuzzy inference system design

Fig.7. shows the flowchart of an adaptive network-based fuzzy inference system (ANFIS), which has five layers of neurons: layer one is the input of the model parameters, called 'premise parameters'; layer two calculates the firing strength of a rule; layer three calculates the normalized firing strength of each rule [19]; layer four maps the output of the membership functions where the parameters in this layer are called 'consequent parameters'; in layer five, the neurons generate the control signal [20].

Combining neural networks and fuzzy logic could enhance the controller's decisions, which are mainly based on the dataset [21]. The collected dataset consists of two inputs and one output – position, velocity and the plate inclination respectively. A joystick is used to balance the ball in different locations while recording the position error, position error rate, and the plate inclination values. The recorded inputs and outputs form all possible combinations to get the ball to the desired position. As part of the training process, the parameters of membership functions are updated using a hybrid optimization method that combines gradient descent and least squares methods [22], [23].

Fig.8. presents the trained data of the fuzzy output, where the root-mean-square error (RMSE) is equal to 0.452, which was obtained after 305 epochs.

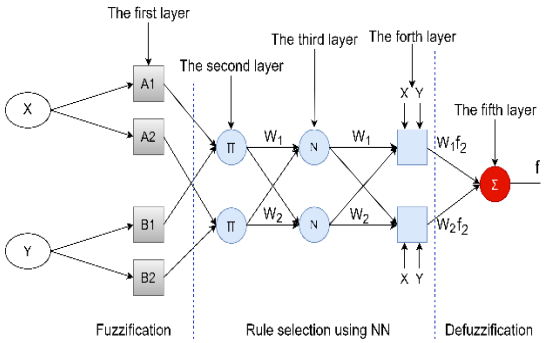

Figure 7. Adaptive neuro-fuzzy inference system block diagram

Figure 8. Training performance

A Sugeno system is chosen for its efficiency in computations [21]. To avoid overfitting due to the high number of parameters, seven membership functions are assigned to each input as well as a trapezoidal membership function type is chosen to improve the control signal accuracy. Fig.9. illustrates the generated membership functions of the position error, which is the difference between the desired position and the actual ball position. Fig.10. describes the generated membership functions of the position error rate, which is the ball velocity.

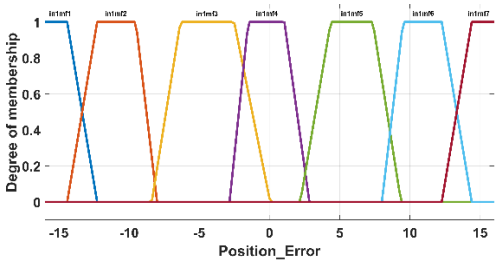

Figure 9. Position error membership functions

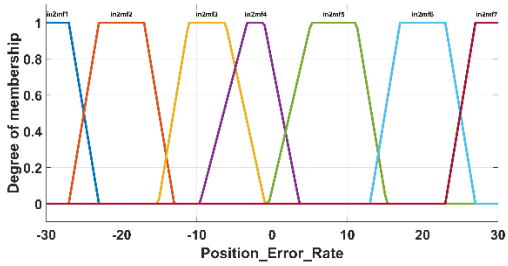

Figure 10. Position error rate membership function

Fig.11 shows the generated surface of the designed controller. The position error and position error rate are the inputs of the controller, while the control signal is limited between -8 and 8. The generated ANFIS structure is portrayed in Fig.12.

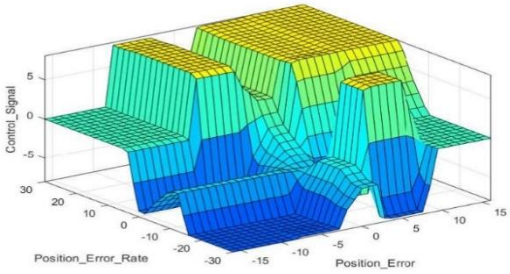

Figure 11. The surface of the fuzzy controller

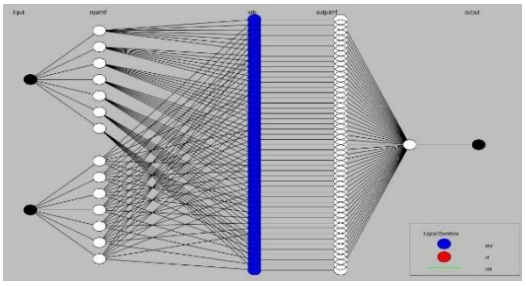

Figure 12. The proposed ANFIS structure

4.2. PID controller

Proportional–integral–derivative is the most used controller worldwide [22]. It takes the ball's position as input and the plate's inclination as output. The controller parameters are tuned using a PID auto-tuning MATLAB toolbox, which are then adjusted experimentally. The proportional gain is Kp = 3.82, the integral gain is Ki = 2.14, and the derivative gain is Kd = 2.91.

4.3. Implementation diagram

4.3.1 Serial processing

In this section, hardware-in-the-loop testing (HIL) is carried out. We start by using the ESP32 microcontroller core 1, which allows serial processing. The controller is implemented using C programming language and image processing is performed on camera. The camera sensor sends X and Y coordinates in pixels to the microcontroller where the pixels are converted to centimeter. Position error and change in error rate are calculated and introduced to the designed controller. After the fuzzification and defuzzification steps, the controller output is fed to the inverse kinematics algorithm. At this level, the servomotor's angles (s1, s2 and s3) are generated and sent to the actuators, as shown in Fig.13.

Two controllers are implemented separately. The first one is responsible for controlling the ball in the X axis while the second one carries out the control of the ball in the Y direction. This increases the complexity of the implemented algorithm.

4.3.2 Decoupling tasks & shared resources

The second test includes core number 0, called the producer, which is responsible for obtaining the X and Y coordinates from the camera and producing the necessary controller output. This is fed into core 1, called the consumer, where inverse kinematics are performed and the servomotor's adequate angles are sent to the actuators, allowing parallel processing, as shown in Fig.14. This type of processing uses the shared resources of roll and pitch angles, which are produced by core 0 and consumed by core 1. Both cores work asynchronously, which means that core 1 is running independently from core 0.

The main idea in this part is the choice of the functions placed in each core. The choice was made on the basis of the algorithmic complexity where the two most greedy functions in terms of resources were placed separately each in a core. The first function that consumes the most resources is the one that contains the execution of the two controllers based on the inputs, membership functions, rules and output of the controller. The second function is the inverse kinematics which requires an exhausting mathematical calculation compared to the resources of the microcontroller.

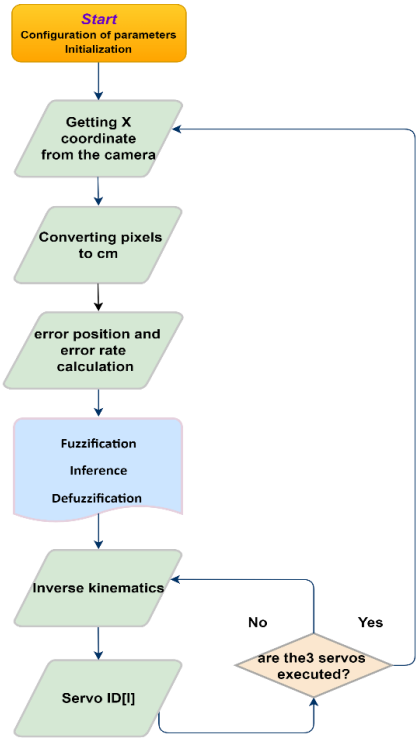

Figure 13. Serial processing control scheme

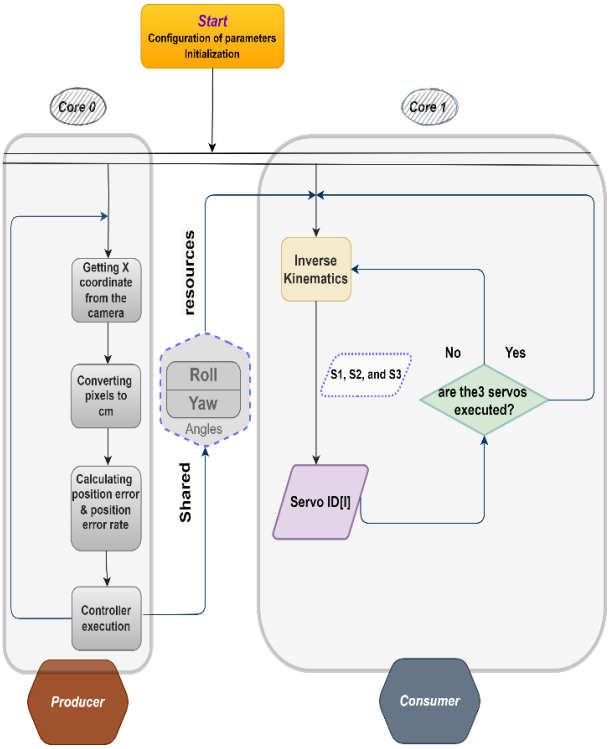

Figure 14. Parallel processing unit

5. RESULTS AND ANALYSIS

5.1. Simulation results

Both controllers were simulated on Simulink using MATLAB software. The scheme presented earlier in Fig.6. was designed and the obtained results are shown in Fig.15.

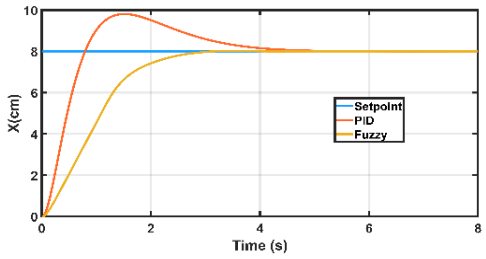

Figure 15. Simulated system response

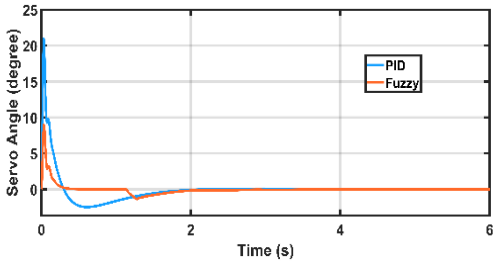

Figure 16. Simulated angle

This figure shows that the PID controller system response achieves the desired position within 0.6092 s of the rise time, with an overshoot of 21.8501% and 4.0311 s of the settling time. By contrast, the ANFIS requires a rise time of 1.5671 s to achieve the set point, with an overshoot of 0.2165% and 2.58 s of settling time. Compared with the PID controller, the ANFIS has a better system response in terms of overshoot and settling time, while the PID controller has a smaller rise-time value.

Fig.16. presents the servomotor angle, where the ANFIS shows a small angle variation compared to the PID controller, leading to minimized oscillations.

5.2. Experimental results

Fig.17a. illustrates the system response of a PID controller, which demonstrates high oscillations as well as a high overshoot rate, with a peak of 24.19 cm and a large tolerance band of steady-state error. Moreover, the system's response shows a rise time of 1.32 s, as presented in Table.2. Fig.17b confirms the long-time adjustment with large servo angles limits.

Fig.17c. demonstrates the system response of the ANFIS when moving the ball to the origin. The ball keeps oscillating around the set point with a high overshoot rate and a short rise time. In addition, the figure

shows that the steady-state error is between a smaller tolerance band when compared to the PID controller. Furthermore, Fig.17d and Table.3 show that the servomotor rotation angles (degree) are limited to a smaller interval when compared to the conventional PID controller response.

From the aforementioned visuals (i.e., Table.2, Table.3 and Fig.18a, 18b, 18c, and Fig.18d), it can be concluded that despite the high peak overshoot demonstrated in 1.59 s, the ANFIS reduces the rise time and increases the plate smoothness while moving the ball to the origin.

Fig.17e. shows the system response of a PID controller using both cores of the microcontroller. It demonstrates a high overshoot rate, with a peak of 26.58 cm and a rise time of 1.42 s. Moreover, the figure depicts that the steady-state error ranges between a small tolerance band of 2 cm and 2.5 cm. However, Fig.17f. exhibits the range of the servomotors angle, which begins in a large limit band and decreases gradually.

Fig.17g. illustrates the system response of the ANFIS using both cores of the ESP32 microcontroller when taking the ball to the origin. It shows a small overshoot rate and has a rise time of 1.44 s. Therefore, the ball reaches the desired point smoothly, with approximately no oscillations and almost no steady-state error. On the other hand, Fig.17h. demonstrates the three servomotors' angle, in which their range of rotation is very limited. This means that the ball reaches its position with smooth movement and minimal effort.

From Fig.17e, 17f, 17g, and 17h, it can be deduced that the ANFIS divided into two cores considerably reduces the overshoot presented in the precedent results, which increases the plate smoothness, stability and guarantees a precise system response when moving the ball to the origin.

Fig.18 illustrates the difference between experimental and simulation results where DPID and DANFIS are the experimental result of PID and ANFIS controllers implemented in both cores of the microcontroller. On the other hand, SPID and SANFIS are the simulation results of ANFIS and PID controllers. It can be seen that the two types of results are different, which is explained by the use of a mathematical model that is not precise enough and contains high uncertainties.

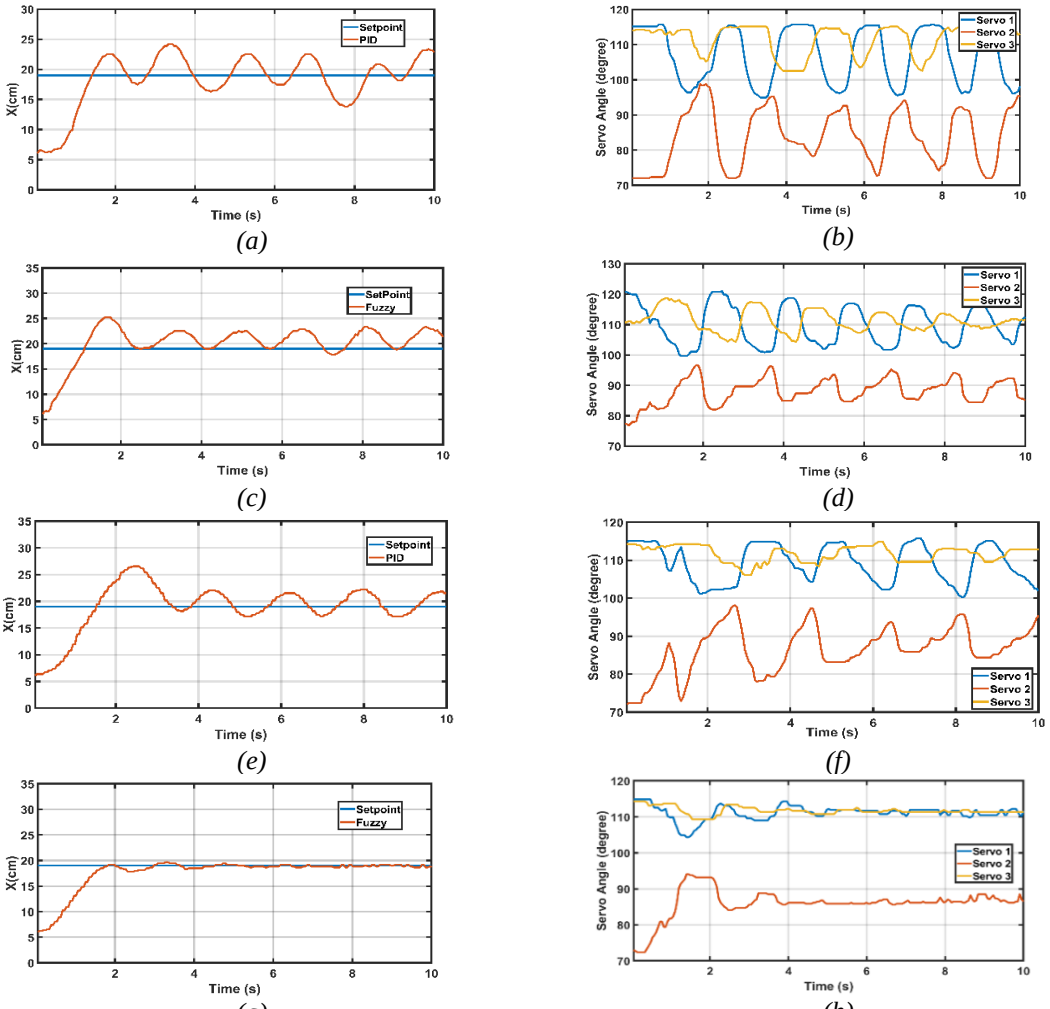

Figure 17. Settling the ball to the origin using PID/ANFIS on dual & single-core

Table 2. Performance metrics comparison between both controllers in single and dual-core

	Single-core			Dual-core		
Controllers	Rise time (s)	S-S error (cm)	Overshoot (%)	Rise time (s)	S-S error (cm)	Overshoot (%)
PID	1.32	-5 to 5	18.4%	1.42	2 to 2.5	40.27%
ANFIS	1.01	0 to 4	36.9%	1.44	0	0.423%

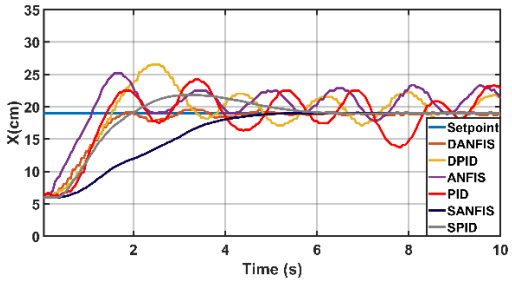

Figure 18.Comparison between simulation and experimental result

In addition to settling the ball to the origin test, rejection disturbance is also evaluated for both controllers using single- and dual-core architectures. Fig.19. shows the performance of each controller for single- and dual-core execution when the ball is moved away from the origin. On one hand, the PID controller demonstrates a high damping ratio when using a single core or a dual core in a smaller interval, which keeps oscillating around the set point. On the other hand, the ANFIS exhibits a good perturbation rejection when using a single core and an excellent result when using a dual core; it rejects perturbation within 4 s with no steady-state error.

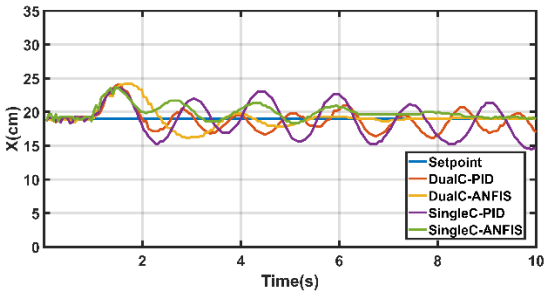

Figure 19. Disturbance rejection using PID and ANFIS on single and dual-core

Table 3. Servo motor excursion limits

Controller	Single-core	Dual-Core
PID	S1=[95-115], S2=[72-98], S3=[102-115]	S1=[110-115], S2=[85-95], S3=[220-115]
ANFIS	S1=[100-120], S2=[100-120], S3=[85-95]	S1=[115-105], S2=[72-93], S3=[115-110]

6. CONCLUSION

In this work, a 3-DOF parallel robot was designed using SolidWorks and manufactured using a CNC machine to control a Ball-On-Plate system. The BPS mathematical model was derived and inverse kinematics were performed. Two types of control techniques were used to evaluate their performance in sequential and parallel asynchronous processing. The first one used a core n°1 of the ESP32 microcontroller, while the second one splits the tasks into two cores: core n°1 was responsible for obtaining the ball's coordinates and executing the proposed controllers, while the second core performed inverse kinematics and sent the rotation angles values to the servomotors. The results demonstrated that the system response could be enhanced by dividing the different tasks into two cores when the used algorithm is complex. This technique allowed us to get a stable and smooth ball movement with a small steady-state error. Furthermore, the disturbance rejection test confirmed the importance of parallel processing when the developed algorithm is complex and resource-consuming. Due to the simplicity of the PID controller algorithm, the results are not enhanced much when compared to the ANFIS, which consumes significant resources. We can conclude that an algorithm that does not consume a lot ofresources (non-complex), can give better results when the platform used is not that powerful. On the other hand, a high complexity algorithm has an asymptotic

behavior whose execution is slow. Therefore, enhancing the efficiency of the implemented algorithm allowed us to improve the obtained results.

REFERENCES

[1] V. Tudić, D. Kralj, J. Hoster, and T. Tropčić, "Design and Implementation of a Ball-Plate Control System and Python Script for Educational Purposes in STEM Technologies," *Sensors*, vol. 22, no. 5, p. 1875, 2022, doi: 10.3390/s22051875.

[2] A. Chevalier, C. Copot, C. M. Ionescu, and R. De Keyser, "Automatic calibration with robust control of a six DoF mechatronic system," *Mechatronics*, vol. 35, pp. 102-108, 2016, doi: 10.1016/j.mechatronics.2016.01.005.

[3] J. M. Rossell, J. Vicente-Rodrigo, J. Rubió-Massegú, and V. Barcons, "An effective strategy of real-time vision-based control for a Stewart platform," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018: IEEE, pp. 75-80.

[4] K. Yaovaja, "Ball balancing on a stewart platform using fuzzy supervisory PID visual servo control," in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018: IEEE, pp. 170-175.

[5] A. Mohammadi and J.-C. Ryu, "Neural network-based PID compensation for nonlinear systems: ball-on-plate example," *International Journal of Dynamics and Control*, vol. 8, no. 1, pp. 178-188, 2020, doi: 10.1007/s40435-018-0480-5.

[6] A. Kassem, H. Haddad, and C. Albitar, "Commparison between different methods of control of ball and plate system with 6dof stewart platform," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 47-52, 2015, doi: 10.1016/j.ifacol.2015.09.158.

[7] A. Negash and N. P. Singh, "Position control and tracking of ball and plate system using fuzzy sliding mode controller," in *Afro-European conference for industrial advancement*, 2015: Springer, pp. 123-132, doi: 10.1007/978-3-319-13572-4_10.

[8] J. Ma, H. Tao, and J. Huang, "Observer integrated backstepping control for a ball and plate system," *International Journal of Dynamics and Control*, vol. 9, no. 1, pp. 141-148, 2021, doi: 10.1007/s40435-020-00629-8.

[9] H. He and J. Chen, "Research on Nonlinear Ball-plate System Based on LTR Method," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020: IEEE, pp. 5243-5248, doi: 10.1109/CCDC49329.2020.9164066.

[10] R. K. Pour, H. Khajvand, and S. A. A. Moosavian, "Fuzzy logic trajectory tracking control of a 3-RRS ball and plate parallel manipulator," in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, 2016: IEEE, pp. 343-348.

[11] E. Okafor, D. Udekwe, Y. Ibrahim, M. Bashir Mu'azu, and E. G. Okafor, "Heuristic and deep reinforcement learning-based PID control of trajectory tracking in a ball-and-plate system," *Journal of Information and Telecommunication*, vol. 5, no. 2, pp. 179-196, 2021, doi: 10.1080/24751839.2020.1833137.

[12] F. I. R. Betancourt, S. M. B. Alarcon, and L. F. A. Velasquez, "Fuzzy and PID controllers applied to ball and plate system," in *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019: IEEE, pp. 1-6.

[13] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665-685, 1993, doi: 10.1109/21.256541.

[14] E. Fabregas, S. Dormido-Canto, and S. Dormido, "Virtual and remote laboratory with the ball and plate system," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9132-9137, 2017, doi: 10.1016/j.ifacol.2017.08.1716.

[15] G. Eleftheriou, L. Doitsidis, Z. Zinonos, and S. A. Chatzichristofis, "A fuzzy rule-based control system for fast line-following robots," in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2020: IEEE, pp. 388-395, doi: 10.1109/DCOSS49796.2020.00068.

[16] F. Faizah, A. Triwiyatno, and R. R. Isnanto, "Fuzzy Logic Implementation on Motion of Tennis Ball Picker Robot," in *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 2021: IEEE, pp. 57-63.

[17] P. Rai and M. Rehman, "ESP32 based smart surveillance system," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019: IEEE, pp. 1-3, doi: 10.1109/ICOMET.2019.8673463.

[18] V. Garg *et al.*, "Anticipatory Postural Adjustments for Balance Control of Ball and Beam System," in *Computational Signal Processing and Analysis*: Springer, 2018, pp. 33-43, doi: 10.1007/978-981-10-8354-9_4.

[19] P. Raja and B. Pahat, "A review of training methods of ANFIS for applications in business and economics," *International Journal of u-and e-Service, Science and Technology*, vol. 9, no. 7, pp. 165-172, 2016.

[20] D. J. Armaghani and P. G. Asteris, "A comparative study of ANN and ANFIS models for the prediction of cement-based mortar materials compressive strength," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4501-4532, 2021.

[21] H. Basarir, M. Elchalakani, and A. Karrech, "The prediction of ultimate pure bending moment of concrete-filled steel tubes by adaptive neuro-fuzzy inference system (ANFIS)," *Neural Computing and Applications*, vol. 31, no. 2, pp. 1239-1252, 2019, doi: 10.1007/s00521-017-3108-3.

[22] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2263-2293, 2019, doi: 10.1007/s10462-017-9610-2.

[23] N. Walia, H. Singh, and A. Sharma, "ANFIS: Adaptive neuro-fuzzy inference system-a survey," *International Journal of Computer Applications*, vol. 123, no. 13, 2015.

[24] P. Bergsten, R. Palm, and D. Driankov, "Observers for Takagi-Sugeno fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 1, pp. 114-121, 2002, doi: 10.1109/3477.979966.
[25] H. S. Purnama, T. Sutikno, S. Alavandar, and A. C. Subrata, "Intelligent control strategies for tuning PID of speed control of DC motor-a review," in *2019 IEEE Conference on Energy Conversion (CENCON)*, 2019: IEEE, pp. 24-30.

BIOGRAPHIES OF AUTHORS

Oussama Hadoune holds a master degree in automation and industrial computing. His research areas include; robotics, real-time implementation, adaptive control and artificial intelligence. He received a number of national awards such the master project which was ranked among the top twenty innovative projects in 2019 by the higher ministry of education in Algeria. This same project won the price of the best third project in the national conference of startup at the university of Oum El-Bouaghi, Algeria. He can be contacted at email: hadouneoussama@gmail.com

Mohamed Benoauret is a professor who received his Bachelors of science and later on his PhD in embedded electronics from the University of Badji Mokhtar Annaba, Algeria. He has supervised and co-supervised more than 20 masters and several Ph.D. students. His research interests include real-time implementation, FPGA, and FFT processors. He can be contacted at email: mohamed.benouaret@gmail.com