Hybrid Adaptive Control for Robotic Arm Using PID-ILC Approach

Martinus Bagus Wicaksono
Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Thailand
martinus_b@mail.rmutt.ac.th

Dechrit Maneetham
Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Thailand
dechrit_m@rmutt.ac.th (corresponding author)

Petrus Sutyasadi
Mechatronics Department, Faculty of Vocational, Sanata Dharma University, Yogyakarta, Indonesia
peter@usd.ac.id

Received: 18 July 2025 | Revised: 15 August 2025, 10 September 2025, and 12 September 2025 | Accepted: 15 September 2025

 DOI: https://doi.org/10.48084/etasr.13495

ABSTRACT

This study presents a hybrid control system that combines Iterative Learning Control (ILC) and Proportional-Integral-Derivative (PID) control to improve the joint control of the SCORBOT ER4u robotic arm. The aim is to improve the accuracy, stability, and efficiency of repeated motion tasks in low-cost instructional robots. PID offers fast response and inherent stability, and ILC progressively refines control performance through repetition. By integrating PID and ILC, the system utilizes their complementary strengths. The controller's performance was evaluated using multiple metrics, including Root Mean Square Error (RMSE), Root Mean Square Percentage Error (RMSPE), Integral of Time-Weighted Absolute Error (ITAE), and Integral of Time-Weighted Squared Error (ITSE), to capture both transient and long-term error characteristics. The hybrid PID-ILC approach reduced steady-state errors by 35% under no-load conditions and by 71% with a 1 kg payload. The system exhibited exceptional stability, with no oscillations or divergence, demonstrating apparent convergence and a bounded response across iterations. These results confirm that the proposed method significantly improves motion precision, reliability, and robustness, making it well-suited for tasks requiring accurate and repetitive control in educational and industrial contexts.

Keywords-PID-ILC; robotic arm control; error reduction; joint control improvement; hybrid control

I. INTRODUCTION

Industrial automation, fundamental to Industry 4.0, enables intelligent manufacturing through the integration of cyber-physical systems, the Internet of Things, and robots [1]. Platforms such as IRoSim play a crucial role in this advancement by integrating CAD-based simulation, planning, and optimization for robotic systems [2]. The importance of robots is highlighted by the fact that they are now a key component of modern industry and perform a wide range of tasks, from precise assembly to material management. Autonomous image-guided robotic systems have been created to perform vision-based tasks in practical applications, enhancing the influence of industrial automation [3].

Industrial automation aims to improve production capacity and speed, reduce output damage, and eliminate worker dangers. To realize these goals, supporting robotic systems is essential in Industry 4.0. Using robotic arms to optimize production processes in manufacturing, food processing [4], and the medical sectors [5] is becoming a feasible solution to achieve these goals. However, the high cost of robotic arms limits their use in many sectors and presents considerable challenges to SMEs [6]. Educational institutions that teach robotics specialists and operators have trouble getting robotic arms due to cost constraints [7]. The high cost of robotic arms limits industry-education cooperation. Thus, cost-effective robotic arms are needed to allow industrial automation in industrial and educational settings [8]. Several researchers have aimed to develop affordable robotic arms [9]. In addition, multiple research efforts have been conducted to create affordable robotic arms that can enhance the teaching experience in universities and vocational schools [10]. Other researchers focused on designing educational robotic arms to develop computational thinking skills in students. In [11], it was observed that existing implementations of instructional robotic arms have not yet addressed their influence on students' computational thinking.

Due to the primary limitation of cost-effectiveness, several researchers used components that are widely accessible and easily obtainable in the market [12]. This situation restricts the functionalities that can be employed on the robotic arm. In addition, this will exhibit worse performance in terms of precision and accuracy of motions compared to industrial robotic arms. For this purpose, several researchers have developed control systems to improve the performance of robot arms, such as a robot arm control system based on the Proportional-Integral-Derivative (PID) algorithm [13]. This study demonstrated that a PID-based control system can improve accuracy and system stability, and minimize steady-state errors. However, the system still has several weaknesses, such as a slight overshoot and steady-state error when controlling downward movement or in the direction of gravitational force [14]. Additionally, manually tuning the PID gain takes a relatively long time and does not necessarily yield optimal performance. Moreover, PID-based control systems are ineffective in handling external disturbances or sudden changes.

To address these limitations, advanced control methods such as adaptive Fixed-Time Sliding Mode Control (FIT-SMC) and adaptive backstepping have been explored. In [15], an FIT-SMC approach was implemented on an anthropomorphic manipulator with friction and unmodeled dynamics. This study further demonstrated that an adaptive backstepping-based sensor and actuator fault-tolerant control method can maintain manipulator performance under sensor and actuator faults, outperforming conventional PID in terms of robustness and adaptability. This study also presented the AUTAREP platform—an open-source, low-cost robotic arm designed for educational and research purposes—that functions as a realistic example of the proposed control approach. AUTAREP demonstrates the growing importance of low-cost robotic systems in schools and Small and Medium-sized Enterprises (SMEs).

Since PID-based control systems alone cannot handle non-linear conditions, they are not suitable for use as robot manipulator controllers. Several studies used additional algorithms along with conventional PID to control robot manipulators, such as in [16], which used a dual-design PID controller to control the manipulator. This control system can more effectively reduce overshooting and save electricity usage than a conventional PID. Nevertheless, this control system is still vulnerable, as it requires additional sample time to enhance its learning and performance. This scenario results in increased latency and hinders real-time system responsiveness. In [17], the social spider algorithm was used in conjunction with fuzzy techniques in a traditional PID controller to optimize the robot arm manipulator. This algorithm can improve robot arm control performance and produce zero steady-state errors. However, this algorithm was only applied to a robot arm model in MATLAB R2021a. In [18], three algorithms were combined with traditional PID to control a SCARA two-limb robot arm in the horizontal plane, namely Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), and Chaos Game Optimization (CGO), with the latter being the best in optimizing the PID controller and controlling the robot arm manipulator.

Several algorithms can be used to control repetitive movements, such as those often performed by robotic arms. The Iterative Learning Control (ILC) algorithm can correct trajectory tracking errors in each iteration based on data collected from the previous iteration [19]. In [20], a hybrid PID-ILC was used to optimize the lower limb exoskeleton joint controller. The results of this research are significant, with steady-state errors being 50% smaller compared to using a traditional PID controller alone. However, in this study, the tested trajectory was only a 2D trajectory with a smaller non-linear factor and used no additional load. Several upper limb hand exoskeleton systems have also been developed to aid or rehabilitate hand movements, in contrast to their lower limb counterparts. The HEXOSYS I system is a multi-degree-of-freedom robotic exoskeleton interface engineered for accurate hand motion support and rehabilitation [21]. Its successor, HEXOSYS II, enhances this concept with a lightweight design prioritizing user comfort and functional performance, signifying progress toward viable hand-wearable robotics [22].

In [23], ILC was applied to reduce vibrations only in the robotic arm due to the elasticity of the robot joints. Experimental results showed that the Variational Mode Decomposition Hilbert-Huang Transform Iterative Learning Control for Elastic Joint (VH-ILC-EJ) method can significantly reduce vibrations, thereby increasing the performance of the entire system. Recent studies have investigated fuzzy logic, adaptive methods, and Fractional-Order PID (FOPID) to enhance control in nonlinear systems. Fuzzy-PID methods improve adaptability in uncertain environments [24], whereas FOPID provides enhanced tuning flexibility and robustness [25]. Adaptive FOPID facilitates real-time gain modification, producing favorable outcomes in intricate dynamic applications. These hybrid methods indicate avenues for prospective improvements in robotic arm manipulation.

These developments indicate the need for more studies on simple hybrid algorithms to enhance the deficiencies of the PID algorithm in controlling a robotic arm, particularly in low-cost educational robotic arms, aimed at improving the precision of its movements within a short timeframe. This study used the SCORBOT ER4u robot arm to lift an object on a 3D path. The base and elbow motions create this 3D trajectory. The system's non-linearity along this trajectory is affected by gravity. Since this kind of system cannot handle external disturbances using only a traditional PID controller, PID was combined with ILC. This hybrid method allows the controller to adapt to changing situations. The controller was used to ensure stability and performance under uncertainties and disturbances. PID-ILC controllers reduce steady-state errors and minimize overshoot during numerous iterations. This study aimed to improve the joint control algorithm of an articulated robotic arm, SCORBOT ER4u, by implementing a hybrid PID-ILC strategy to ensure precision and stability even when the elbow experiences load variations (disturbances) and minimize any overshoot and steady-state errors as the system attempts to track a 3D reference trajectory accurately.

II. METHODOLOGY

A. Experimental Design

The subject of investigation is the robotic arm SCORBOT ER4u, with technical specifications shown in Table I, which is used due to its mechanical benefits. These advantages arise from the gearbox system that links the motor to the joint, which functions as an arm drive or a link utilizing a timing belt, as seen in Figure 1. This design enables the consolidation of all motors as actuators in a single location, at the base. Other variants of articulated robotic arms have motors at each joint, resulting in increased strain on each link connected to the motorized joint.

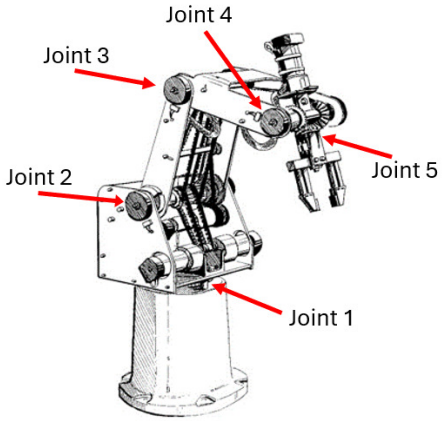

Fig. 1. SCORBOT ER4u robotic arm.

For this investigation, joint 1 and joint 3 are adjusted to match a pre-determined trajectory. The path adheres to a modified sinusoidal function described by:

$$(8000 \times sin(x) + 100) + (1000 \times cos(2x) + 100) \quad (1)$$

where x is updated to vary the setpoint over time, creating a modified sinusoidal trajectory.

$$x = x + \left(\left(\frac{\pi}{2} \times 10^{-2}\right) \times 3.855\right) \quad (2)$$

x involves several pulses, which are used to calculate the setpoint on the trajectory. A sinusoidal setpoint is generated, and the variable x is incremented. To convert the x value into an angular value for each joint, refer to Table II.

TABLE I. SCORBOT ER4U TECHNICAL SPECIFICATIONS

Specification	Value
Model	SCORBOT ER4u
Degrees of Freedom (DoF)	5 (all are revolute joints)
Payload capacity	1 Kg
Joint 1 Range motion	± 165°
Joint 2 Range motion	+ 130° / -35°
Joint 3 Range motion	± 130°
Joint 4 Range motion	± 180°
Joint 5 Range motion	± 270°
Drive system	DC Motor with timing belt + gears
Control system	Arduino Mega + BTS7960 Driver
Feedback	Rotary encoders on all joints
Power supply	12V DC, 10A

Figure 2 displays the path that the robot arm's tip will follow. The trajectory is the result of the combined motions of joint 1 and joint 3. Figure 2 displays the trajectory for moving joint 1 in the top view, while the front view image guides the movement of joint 3.

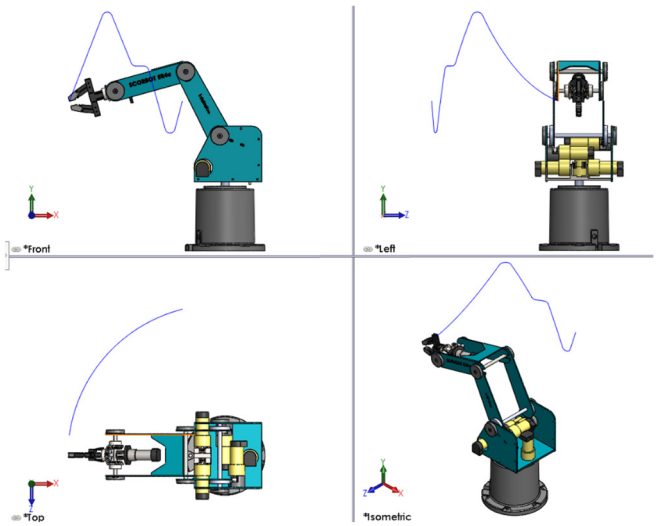

Fig. 2. Robotic arm's 3D trajectory.

Figure 2 shows the combined movement from an isometric perspective. The following actions were taken for the experimental scenario:

1. Joints 1 and 3 follow the setpoint trajectory and are controlled by the PID algorithm without external force.
2. Joints 1 and 3 follow the setpoint trajectory and are controlled by the PID-ILC algorithm without external force.
3. Joints 1 and 3 move with a 1 kg external force, following the setpoint trajectory and controlled by the PID algorithm.
4. Joints 1 and 3 follow the setpoint trajectory and are controlled by the PID-ILC algorithm with an external force of 1 kg.

The data obtained from the motor's responses while following the predetermined setpoint trajectory are collected and then analyzed.

Several assumptions were established in the design and execution of this investigation to ensure the feasibility and consistency of the control evaluation. Robotic joints were initially presumed to function autonomously, devoid of mechanical backlash or substantial frictional influences, facilitating streamlined modeling and controller development. The mechanical design of SCORBOT ER4u, with base-mounted actuators, naturally reduces coupling effects, although dynamic coupling between joints was not explicitly modeled. This makes it easier to establish control and reduces interference between joints in the conditions tested.

TABLE II. NUMBER OF PULSES PER DEGREE OF JOINTS

Joint	#pulses per degree
1	125
2	115
3	115
4	65
5	65

The control experiments were performed in a steady, noise-free laboratory setting with a controlled power source, ensuring consistent voltage and reducing external interference. The impact of motor thermal dynamics and actuator saturation was considered insignificant during the brief duration of each trial. External disturbances were mainly modeled by delivering a constant load of 1 kg to the end effector, reflecting typical gravitational forces seen in practical applications. Although these assumptions are valid for a controlled experimental framework, they may restrict direct scaling to more complicated industrial settings, where unmodeled dynamics, noise, and hardware limitations are more pronounced. A future study will examine addressing these factors through adaptive or observer-based control techniques.

B. Kinematics

Kinematics is the study of the motion of a body or a group of bodies, disregarding their mass or the forces acting upon them [26]. The controlled object is the SCORBOT ER4u vertical articulated manipulator with five revolute joints. Figure 3 shows the links and joints of the robotic arm. A kinematic diagram was created to simplify the mathematical representation of the SCORBOT ER4u robot arm, as shown in Figure 4. The kinematic diagram was utilized for determining the Denavit-Hartenberg (D-H) parameters, shown in Table III.

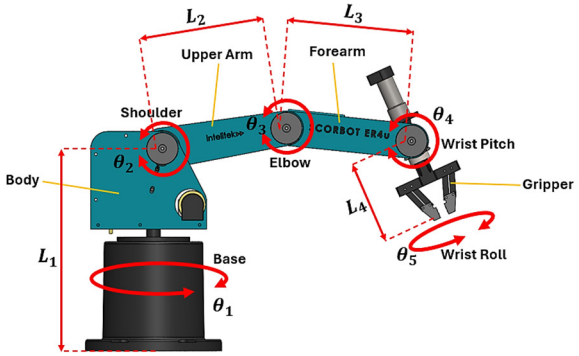

Fig. 3. SCORBOT ER4u links and joints.

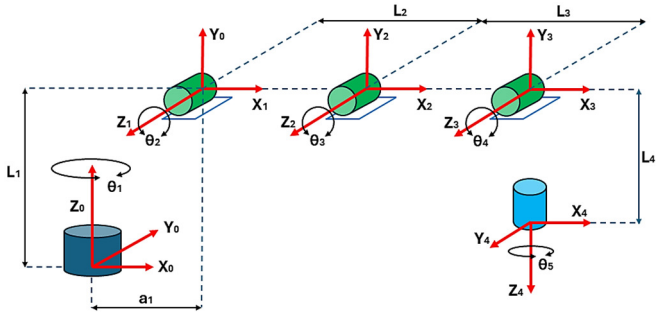

Fig. 4. Kinematic diagram of SCORBOT ER4u.

TABLE III. DENAVIT-HARTENBERG PARAMETERS

Joint n	θ_n (deg)	α_n (deg)	a_n (cm)	d_n (cm)
1	θ_1	π/2	$a_1 = 1.5$	$L_1 = 35$
2	θ_2	0	$L_2 = 22$	0
3	θ_3	0	$L_3 = 22$	0
4	θ_4	π/2	0	0
5	θ_5	0	0	$L_4 = 14$

The D-H parameters simplify the mathematical model of robotic manipulators [27]. These characteristics can precisely define the position and orientation of every joint and link, improving the process of analyzing and controlling robot motions. D-H parameters involve four variables: θ (joint angle), α (link twist), r (link length), and d (link offset). The first pair of variables belongs to rotation, whereas the last pair relates to displacement. Figure 4 shows the kinematic diagram of SCORBOT ER4u, which is related to Table II, for θ_n being the angle from X_{n-1} to X_n about Z_{n-1}. Angle α_n can be calculated from the rotation of Z_{n-1} to Z_n about X_n. Distance a_n is a measure from origin frame $n-1$ to origin frame n along the X_n direction. The last is the distance d_n, measured from X_{n-1} to X_n along the Z_{n-1} direction. The forward kinematics are obtained using a 4×4 homogeneous transformation matrix [28]:

$$T_{n-1,n} = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} = \begin{bmatrix} cos\theta_n & -sin\theta_n.cos\alpha_n & sin\theta_n.sin\alpha_n & a_n.cos\theta_n \\ sin\theta_n & cos\alpha_n.cos\theta_n & -cos\theta_n.sin\alpha_n & a_n.sin\theta_n \\ 0 & sin\alpha_n & cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In this context, R denotes the upper left 3×3 submatrix, which shows how to rotate from frame $n-1$ to frame n, while T in the top right signifies the 3×1 submatrix, showing the change in position (or translation) from frame $n-1$ to frame n [29]. The substitution of D-H parameters into (3) provides the transformation matrix for each link.

$$T_{0,1} = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & a_1 cos\theta_1 \\ sin\theta_1 & 0 & -cos\theta_1 & a_1 sin\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_{1,2} = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & a_2 cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & a_2 sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_{2,3} = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3 cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3 sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$T_{3,4} = \begin{bmatrix} cos\theta_4 & 0 & sin\theta_4 & 0 \\ sin\theta_4 & 0 & -cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$T_{4,5} = \begin{bmatrix} cos\theta_5 & -sin\theta_5 & 0 & 0 \\ sin\theta_5 & cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Hence, the homogeneous transformation matrix used for transforming the coordinates of position and orientation of every joint and link could be expressed by (9), (10), and (11) [30]:

$$T_{0,5} = T_{0,1} * T_{1,2} * T_{2,3} * T_{3,4} * T_{4,5} \quad (9)$$

$$= \begin{bmatrix} M_{3\times3} & P_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \quad (10)$$

$$T_{0,5} = \begin{bmatrix} M_1 & M_2 & M_3 & P_X \\ M_4 & M_5 & M_6 & P_Y \\ M_7 & M_8 & M_9 & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where:

$$M_1 = cos\theta_1 cos\theta_5 cos(\theta_2 + \theta_3 + \theta_4)$$

$$M_2 = cos\theta_5 sin\theta_1 - sin\theta_5 cos\theta_1 cos\theta_2 cos(\theta_3 + \theta_4)$$

$$M_3 = cos\theta_1 sin(\theta_2 + \theta_3 + \theta_4)$$

$$M_4 = -cos\theta_1 sin\theta_5 + cos\theta_5 cos\theta_2 sin\theta_1 cos(\theta_3 + \theta_4) - sin\theta_1 sin\theta_2 sin(\theta_3 + \theta_4)$$

$$M_5 = -cos\theta_1 cos\theta_5 - sin\theta_5 sin\theta_1 cos\theta_2 cos(\theta_3 + \theta_4) - sin\theta_2 sin(\theta_3 + \theta_4)$$

$$M_6 = sin\theta_1 sin(\theta_3 + \theta_4)\,[cos\theta_2 - sin\theta_2]$$

$$M_7 = cos\theta_5 \left(cos\theta_2 sin(\theta_3 + \theta_4) + sin\theta_2 cos(\theta_3 - \theta_4)\right)$$

$$M_8 = -sin\theta_5 sin(\theta_3 + \theta_4 + \theta_5)$$

$$M_9 = -cos(\theta_2 + \theta_3 + \theta_4)$$

$$P_X = cos\theta_1 (a_3 cos(\theta_3 + \theta_2) + a_2 cos\theta_2 + d_5 cos\theta_2 sin(\theta_3 + \theta_4) - sin\theta_2 cos(\theta_3 + \theta_4) + r_1)$$

$$P_Y = sin\theta_1 \begin{pmatrix} a_3 cos(\theta_2 + \theta_3) + a_2 cos\theta_2 + \\ d_5 sin(\theta_2 + \theta_3 + \theta_4) + a_1 \end{pmatrix}$$

$$P_Z = a_3 sin(\theta_2 + \theta_3) + d_1 - d_5 cos\theta_2 cos(\theta_3 + \theta_4) + sin\theta_2 sin(\theta_3 + \theta_4) + a_2 sin\theta_2$$

The accessible workspace of the SCORBOT ER4u was calculated by applying the D-H parameters from Table III and the joint Range of Motion (RoM) values from Table I, sweeping joint angles θ_1-θ_5 over their mechanical limits with a resolution of 1°. Forward kinematics were assessed utilizing the homogeneous transformation $T_{0,5}$ (9-11) to determine the end-effector location for each configuration. The outcomes are shown in Figure 5. The 3D graphic illustrates a roughly spherical envelope defined by the collective extension of links L_1–L_4 and the constraints of the joint limits. The side view verifies a vertical range of around -15 cm to +90 cm in the Z axis, while the superior view emphasizes the circular symmetry in the horizontal plane, with slight discontinuities due to joint motion limitations. For the visualization of the positional workspace, θ_4 and θ_5 were fixed, as their movements mostly influence orientation rather than position.

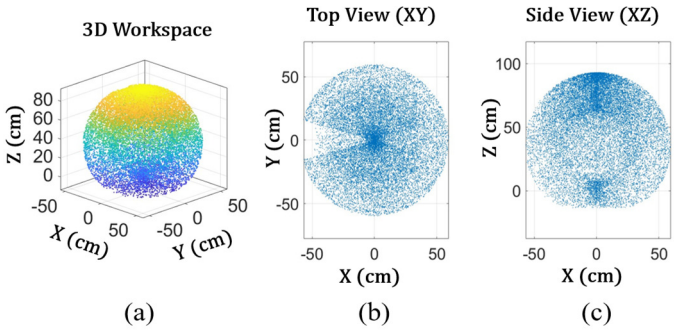

Fig. 5. Workspace of the SCORBOT ER4u derived from the D-H model: (a) 3D point cloud, (b) XY plane top view, (c) XZ plane side view.

These graphs confirm that the kinematic model and ROM data precisely define the operational reach of the SCORBOT ER4u, offering a unique geometric framework for following control and trajectory planning.

C. Motor Driving System and Microcontroller

In the experiments, the electronics of the robot consist of an Arduino MEGA microcontroller, a BTS7960 H-Bridge DC Motor Driver, a 12V-10A power supply, and a DC Motor Pittman GM9213E922 with a rotary encoder. Figure 6 shows the block diagram of the electronics. The components for the system controller were chosen based on several criteria, including component availability, programming simplicity, and cost-effectiveness. The Arduino MEGA is used to generate the trajectory and control algorithm, specifically the PID-ILC. Every motor is used to manipulate each individual joint. The motors are fitted with a rotary encoder to ascertain the precise location of each motor as it traverses the current trajectory. The precise positional reading will provide feedback for the robot control system. The robot arm will be used as an instructional instrument for students to develop their skills and expertise.

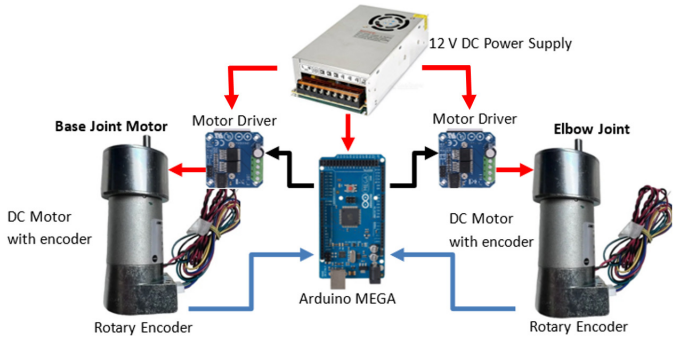

Fig. 6. Block diagram of the electronic system.

D. PID-ILC Control

In robotic systems, PID control is utilized to minimize the discrepancy between the reference and the measured estimate [31]. The PID controller, illustrated in Figure 7, is designed to regulate a process variable, denoted as $y_{Out}(t)$, to a desired setpoint $r_{in}(t)$ by adjusting to compensate for any changes $e_k(t)$ that may arise. The system creates control signals $U_C(t)$ to govern the system and achieve the setpoint value with speed and accuracy.

There are three gain parameters in PID control, namely Proportional (Ψ_p), Integral (Ψ_i), and Derivative (Ψ_d). Proportional control output matches the error value exactly. The error $e_k(t)$ is the difference between the measured process variable $y_{Out}(t)$ and the intended setpoint $r_{in}(t)$. The higher Ψ_p values increase system responsiveness but may cause overshooting and instability. The Ψ_i control component examines globally the previous errors. Integrating the error across time reduces or eliminates steady-state error. The integral action raises the integral term if the error continues, moving the process variable closer to the setpoint. The Ψ_d control component forecasts the error direction based on the rate of change. Damping lowers the system's deviation from the target value. The Ψ_d action estimates future errors and adjusts control output. The combination of the controlled output is the overall sum of the proportional, integral, and derivative gains:

$$U_c(t) = \Psi_p \,.\, e_k(t) + \Psi_i \,.\, \int_0^t e_k(t)dt + \Psi_d \,.\, \frac{de_k(t)}{dt} \quad (12)$$

where $U_C(t)$ is the output of the PID controller, Ψ_p is the proportional gain, $e_k(t)$ is the difference between the setpoint and the process variable, Ψ_i is the integral gain, and Ψ_d is the derivative gain.

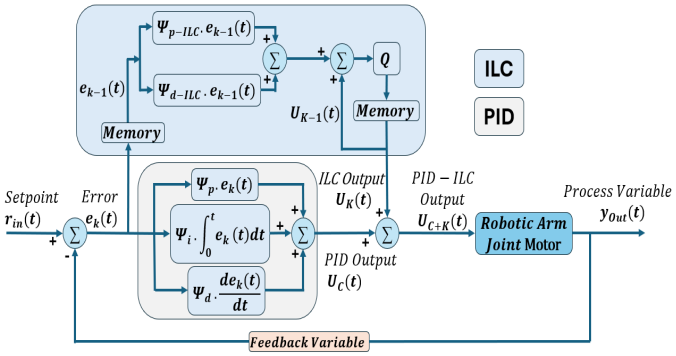

Fig. 7. Block diagram of the hybrid adaptive PID-ILC applied to a robotic arm.

Several PID tuning approaches may improve control system responsiveness. Standard tuning methods include manual tuning, using expertise and trial-and-error to modify parameters. The Ziegler-Nichols heuristic approach analyses the system's response to specific tests to provide initial parameter values. Software-based tuning optimizes settings using algorithms and simulations. This PID tuning sets proportional, integral, and derivative PID control parameters. These factors impact system stability and resilience. Setpoint achievement is a reference variable for system performance. Several studies have used PID [32] or hybrid PID [33] as the system controller.

Due to its several limitations, such as the inability to regulate non-linear systems, PID control is not suitable for such applications. To address this issue, it is necessary to use an additional control method that may be integrated with PID control to mitigate its limitations. One such algorithm is ILC, which is an adaptive control approach that enables the system to gain learning from its prior activities and consistently improve the performance of repetitive tasks [34]. The ILC controller output $U_k(t)$, as illustrated in Figure 7, is used to eradicate tracking inaccuracy by utilizing information from previous data. If a control system consistently produces the same tracking error when given the same command, the error $e_{k-1}(t)$ and the control input signals from the prior iteration must be adjusted to provide a more appropriate control input signal that reduces the tracking error in this execution. When using the ILC, certain scenarios must fulfill the following criteria:

- The trajectory should include a repeated job.
- The repeated trajectory should have identical starting and ending points.
- Before the ILC is implemented, the system must be stable.

The control signal of ILC is derived from

$$U_k(t) = U_{k-1}(t) + \Psi_{p-ILC}.e_{k-1}(t) + \Psi_{d-ILC}.e_{k-1}(t) \quad (13)$$

where $U_k(t)$ is the current ILC control signal, U_{k-1} is the previous control signal, $e_{k-1}(t)$ is the previous error, Ψ_{p-ILC} is the proportional ILC gain, Ψ_{d-ILC} is the derivative ILC gain, and k is the iteration number.

The robotic arm joint motor will be controlled by the PID-ILC control algorithm, which is a combination of the outputs of the ILC and PID control algorithms. Figure 6 illustrates how this process works. PID-ILC may be expressed as the summation of (12) and (13):

$$U_{C+k}(t) = U_C(t) + U_k(t) \quad (14)$$

Thus, the PID-ILC can be described as:

$$U_{C+k}(t) = \left[\Psi_p.e_k(t) + \Psi_i.\int_0^t e_k(t)dt + \Psi_d.\frac{de_k(t)}{dt}\right] + \left[U_{k-1} + \Psi_{p-ILC}.e_{k-1}(t) + \Psi_{d-ILC}.e_{k-1}(t)\right] \quad (15)$$

In the initial condition, the value of the process variable is zero because the value of the setpoint is also zero. When the system is working, the output of the PID-ILC will be used to control the robotic arm joint's motor, as shown in Figure 7. In this case, the resulting process variable is expected to follow the setpoint input to the system. In other words, this PID-ILC ensures that the process variable $y_{Out}(t)$ has the same value as the setpoint $r_{in}(t)$ in each process.

To assess the tracking accuracy of control systems, numerous studies have employed performance metrics including Root Mean Square Error (RMSE) and Root Mean Square Percentage Error (RMSPE). For example, in [35], it was shown how to utilize RMSE to assess how well an ILC performed on an articulated robotic arm. These errors are represented by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_{i,obs} - Y_{i,set}\right)^2}{n}} \quad (16)$$

$$RMSPE = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{\left(Y_{i,obs}-Y_{i,set}\right)^2}{Y_{i,obs}}\right)\times 100}{n}} \quad (17)$$

where n is the number of data and $Y_{i,set}$ and $Y_{i,obs}$ are the individual setpoint and observed values, respectively. RMSE is a common way to measure model errors from quantitative data predictions. RMSE is used to evaluate the size of the variations of data points from the linear regression line or the distribution of data surrounding the linear regression line. RMSE quantifies the immediate effectiveness of a model by enabling a point-by-point assessment of the actual difference between the desired and observed values. Therefore, the control algorithm is more effective when the RMSE and RMSPE values are modest.

This study also employs time-weighted performance indices, Integral of Time-weighted Absolute Error (ITAE) and Integral of Time-weighted Squared Error (ITSE), to provide a more comprehensive assessment of control performance, considering both the magnitude and duration of errors over time [36]. These metrics are calculated using

$$ITAE = \int_0^T t|e(t)|dt \quad (18)$$

$$ITSE = \int_0^T t\, e^2(t)dt \quad (19)$$

where $e(t) = Y_{set}(t) - Y_{obs}(t)$ is the instantaneous tracking error and T is the observation period. Unlike RMSE and RMSPE, which assess the average error magnitude, ITAE and ITSE impose greater penalties on errors that persist over extended periods, thereby prioritizing the reduction of steady-state deviations and enhancement of long-term accuracy. This renders them especially pertinent for robotic arm control, where both transient reaction and final position precision are essential. Decreased ITAE and ITSE values signify expedited settling times and reduced long-term error, which are necessary for applications requiring accurate motion tracking under fluctuating load conditions.

III. RESULTS AND DISCUSSION

A. PID and PID-ILC Tuning

This study started the robotic arm control system using PID control, adjusting the gain parameters (Ψ_p, Ψ_i, Ψ_d) to enhance the system's response to the setpoint. The Ziegler-Nichols tuning method tunes PID controller settings, finding the ideal gain value, followed by trial-and-error fine-tuning. The starting values of Ψ_i and Ψ_d are zero. Additionally, Ψ_p is gradually increased until the system achieves steady and continuous oscillation, which is caused by the system's aggressive response to a significant correction signal. The system reacts quickly to reduce the error, but its robust response pushes it over the preset threshold. The crucial Ψ_p (K_C) gain value is calculated at this moment. The critical period (T_C) can be determined from the oscillations that take place, as shown in Figure 8. The close-up image details the oscillation from the 5th to the 6th second. K_C is 11, and the data collection sampling rate is 10 Hz, corresponding to 10 data points per second. Two waves are observed within one second, resulting in a T_C of 500 ms.

Moreover, the gain value of each variable (Ψ_p, Ψ_i, and Ψ_d) may be determined using the method shown in Table IV. According to Table IV, the gain values for Ψ_p, Ψ_i, and Ψ_d are 6.6, 26.4, and 0.4125, respectively. The gain values are applied to the system, and the response of the system is shown in Figure 9 by the green line. The fine-tuning step is significant for enhancing the system's responsiveness, yielding gain values Ψ_p, Ψ_i, and Ψ_d of 6.6, 0.01, and 0.4125, respectively. A more stable system response is achieved under this scenario, as seen by the red line in Figure 8.

Once the PID control has achieved a steady response, the ILC algorithm can be used as a hybrid control method on the SCORBOT ER4u. The values for the parameters Ψ_{p-ILC} and Ψ_{d-ILC} are 2.5 and 0.1, respectively.

TABLE IV. ZIEGLER-NICHOLS TUNING FORMULA

Controller type	Ψ_p	Ψ_i	Ψ_d
P	$0.5 \times K_C$		
PI	$0.45 \times K_C$	$\frac{1.2 \times \Psi_p}{T_C}$	
PID	$0.6 \times K_C$	$\frac{2 \times \Psi_p}{T_C}$	$\frac{T_C \times \Psi_p}{8}$

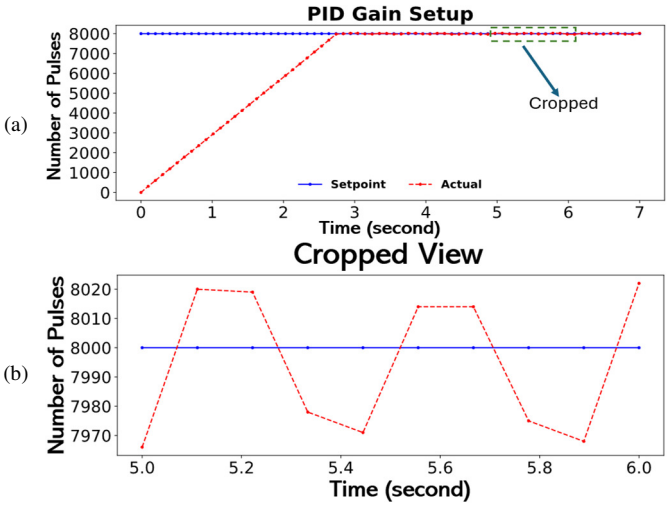

Fig. 8. (a) Response system at critical K_C with continuous oscillation, (b) Zoom out of the cropped view.

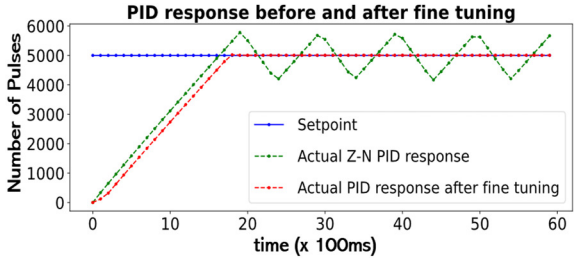

Fig. 9. PID response before and after fine tuning.

The experiment was conducted using the microcontroller as a means of controlling the device. The experimental procedures and data gathering were conducted in accordance with the scenario explained in the experimental design.

B. Results of System Movements Controlled by PID Only and PID-ILC

This study used a modified sine wave trajectory to analyze the effectiveness of joint control. The used joints are joint 1 and joint 3. Both joints were synchronized to move in unison along a predetermined trajectory, creating a 3D trajectory. Figures 10 and 12 show the performance of joint 3 in various test settings. Multiple stages were used to implement the control algorithm. In the first case, the system was controlled using the PID control algorithm without ILC. There was no load attached to the end of the link driven by joint 3. Figure 10 illustrates the system's performance using PID control when the motor-controlled link at joint 3 was manipulated to track a trajectory without any additional load at the end of the link. The trajectory can be tracked correctly, although errors still occur, especially when the system experiences rapid changes in the direction of movement. Significant changes in the direction of movement from bottom to top are due to the significant moment of inertia in the system. The system could maintain stability in the presence of steady-state errors ranging from 0 to 0.514°.

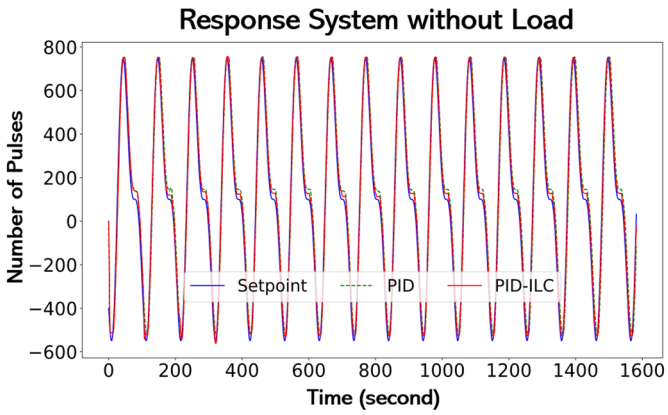

Fig. 10. Response system without load using PID and PID-ILC.

Once the PID control provided a steady response, the Iterative Learning Control (ILC) process was launched, and the system's response is illustrated in Figure 9. The system remains to use the same scenario, where the load at the end of the link is not lifted, and the gain parameter in the PID control remains unchanged. The use of the PID-ILC control algorithm shows a substantial performance improvement. The current steady-state error will gradually decrease and reach the setpoint value because of the ILC algorithm's correction of the current error based on the prior error. By the 15th iteration, the system will have more stability, and the steady-state error may be reduced to 0–0.319°. Figure 10 shows the performance comparison of the two algorithms in controlling joint 3 with no-load conditions. The system's response using PID-ILC is closer to the given setpoint than when using the PID algorithm unaided to control the system.

In the second case, a force is exerted on the end of the link, which is moved by joint 3, as illustrated in Figure 11. The load used is 1 kg, as specified by the maximum load capacity stated in the SCORBOT ER 4U User Manual [37]. The objective of using loads is to evaluate the system's response and performance while lifting the load during work changes. Similarly to the first case, the control technique used in the first experiment is PID, with the same gain variable as in the first experiment.

Fig. 11. SCORBOT ER 4U lifting a 1 kg load.

Figure 12 shows that the PID controller does not exhibit satisfactory performance in the motion of joint 3 when used to lift the load. This is shown by the occurrence of a somewhat substantial steady-state error. The system exhibits stability; however, it has steady-state errors that vary between 0 and 1.2°. This shows that the PID controller is not suitable for nonlinear systems [38]. Although classical PID controllers can maintain system stability, their ability to eliminate steady-state errors in nonlinear systems is minimal. In [39], it was shown that PID can guarantee finite—but not-zero—steady-state tracking errors affected by disturbances and reference dynamics, which is consistent with the observed error of 0–1.2° at joint 3. Recent experiments on flexible robot manipulators have shown that robust tracking control of flexible manipulators using hybrid backstepping nonlinear reduced-order active disturbance rejection can provide stability, but does not eliminate tracking errors under nonlinear uncertainties [40].

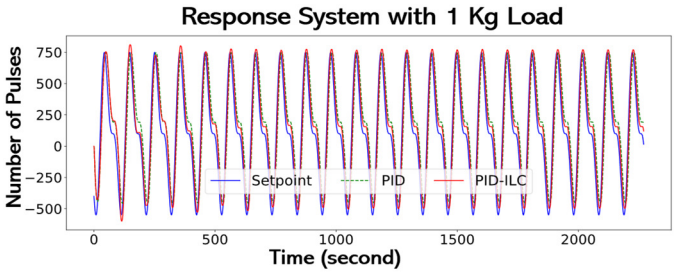

Fig. 12. Response system with 1 kg load using PID and PID-ILC.

PID-ILC control was used in the final experiment to improve the system performance. Although maintaining the same gain and load controllers, the system reacts differently. Figure 12 shows that after 20 iterations, the system responds well. The performance of PID-ILC is indicated by the steady-state error decreasing until it converges at 0 to 0.305°. In contrast to the first experiment, which needed only 15 iterations to obtain its optimum result, the second experiment took 20 iterations to reach its ideal results. The variation in the number of repetitions is attributable to the differing circumstances, namely the additional load applied in the second experiment. This load initially results in more inaccuracies in the robotic

arm's movement. The PID-ILC algorithm corrects the error and achieves optimal performance after the 20th iteration. This means that if this algorithm is utilized for low-cost robotic arms, it will create only 20 defective products in mass production, which is very small and suitable for mass production. 0.305° can be considered an insignificant error value. Figure 11 shows that the hybrid PID-ILC control approach increases the system's capacity to manage external disturbances and uncertainty. The results reveal that the actual and planned trajectories are comparable. This supports the basic notion that supplementary algorithms are necessary with PID to regulate joints in robotic arms and manage external disturbances and uncertainties that may occur during operation.

Figures 10 and 12 show how the PID and PID-ILC controllers behave over time. To quantitatively assess performance, essential transient and steady-state characteristics were derived from the responses of the PID and PID-ILC controllers, as detailed in Table V. The parameters are rising time (the time taken to get from 10% to 90% of the setpoint), peak time, percentage overshoot, settling time (the first time it enters the ±5% region of the steady-state value), and steady-state error. The PID-ILC controller had a rise time of about 0.10 seconds, a peak time of 34.4 seconds, an overshoot of 166.187%, and a settling time of 25.1 seconds, with very little steady-state error. The rising time stayed the same at 0.10 s with a 1 kg load, while the peak time was 34.1 s, the overshoot was 171.693%, and the settling time was 35.6 s. These results show that the suggested hybrid PID-ILC controller keeps the transient behavior stable even with the extra load, while also effectively reducing steady-state error in both cases. The PID-ILC method showed much faster convergence and a smaller final error than the standard PID controller, demonstrating that the PID-ILC strategy is better. It also shows that using iterative learning can help reduce the effects of load disturbances and improve tracking accuracy.

TABLE V. STEP RESPONSE METRICS

Metric	PID		PID-ILC	
	Without load	With 1 kg load	Without load	With 1 kg load
Rise time (s)	0.1	0.1	0.1	0.1
Peak time (s)	34.4	34.4	34.4	34.1
Overshoot (%)	166.605	171.633	166.187	171.693
Settling time (s)	32.7	34	25.1	35.6
Steady-state error (pulses)	74.7	73.03	74.52	60.05

Table VI presents a quantitative study that compares the responses of PID and PID-ILC using RMSE and RMSPE. In the no-load experiment, the comparison of steady-state errors when using the PID control algorithm and the PID-ILC control algorithm decreased by 35%. In the experiment using a load of 1 kg, the PID-ILC control algorithm decreased steady-state error by 71% compared to when using only the PID control algorithm. The disparity in RMSE values between PID and PID-ILC under no-load situations is not substantial, indicating that the PID control settings used are optimal, as seen by the minimal disparity in RMSE values between the two controller types. However, this circumstance alters when the robotic arm is used to traverse a predetermined trajectory by hoisting a load. The RMSE value almost doubles when the system alone employs PID control. Meanwhile, PID-ILC control enhances system performance by successfully reducing current steady-state error. The experimental results are consistent with the initial hypothesis that the PID-ILC algorithm, employed to regulate the joints of low-cost articulated robotic arms, will enhance steady-state errors in the presence of some disturbances in the work process. To further measure control performance, ITAE and ITSE were used, which show longer-lasting errors and that the controller can reduce sustained deviation. Table VI shows that the PID-ILC controller had lower ITAE and ITSE values than the PID when there was no load.

TABLE VI. ERROR COMPARISON OF PID AND PID-ILC RESPONSES

Load	Without load		With 1 Kg load	
Control Type	PID	PID-ILC	PID	PID-ILC
RMSE	66.06	43.39	128.74	64.85
RMSPE	5.10%	3.28%	12.48%	3.61%
ITAE	8674.64	4651.29	6112.23	13933.05
ITAE%	4.94%	4.96%	15.33%	5.57%
ITSE	707800.22	198006.12	887156.82	1244821.99
ITSE%	1.01%	0.28%	2.97%	0.66%

PID-ILC had a greater ITAE than PID when the load was 1 kg, showing that it improved steady-state accuracy but had a bigger transient departure at first because of load dynamics. The PID-ILC's ITSE percentage became better with each iteration, which shows that it can adapt. The results suggest that the hybrid controller works, although it could be better if it could respond to heavier loads.

Previous works have noticed some limitations of conventional PID controllers: they achieve stable motion but undergo steady-state error in disturbances [14]. Hybrid PID-ILC controllers perform better than classical PID, as indicated in [20], which showed improvements in exoskeleton joints; however, this study extends validation to a 3D robotic arm trajectory with payload disturbances, where a 35% error reduction was reported in the no-load case and 71% in the loaded case. Moreover, although in [32] it was reported that ILC can eliminate joint vibrations, these results indicate that this hybrid PID-ILC framework can also alleviate payload-tracking errors with convergence within 20 iterations. These comparisons experimentally validate PID-ILC in real hardware scenarios involving nonlinear conditions that are relevant educationally and for SMEs.

Performance analysis demonstrated that the hybrid PID-ILC controller significantly improves tracking accuracy and reduces steady-state error under both unloaded and loaded conditions. The proposed hybrid PID-ILC controller maintained stable behavior across all test conditions, with decreasing error over iterations. The system did not show divergence or oscillation, indicating good convergence and a bounded response. This confirms the controller's practical stability, especially in repeated motion tasks with external disturbances. Although the proposed control algorithm performs well under controlled conditions with repeated motion, it has some limitations. The system assumes consistent

trajectories and steady environmental conditions, which may not accurately represent real-world scenarios characterized by random disturbances or non-periodic inputs. The controller also does not compensate directly for joint coupling, actuator saturation, or changes in system characteristics over time. Future research will focus on integrating adaptive mechanisms that can auto-tune control gains in response to real-time feedback, alongside observer-based methods to estimate and mitigate unmeasured disturbances and nonlinear dynamics.

IV. CONCLUSION

The proposed PID-ILC hybrid scheme was integrated into the SCORBOT ER4u robotic arm to accurately and stably track complex 3D trajectories. In this study, the PID control system allowed the robot to move smoothly, with a steady-state error of 0.514° for unloaded motion and 1.2° for a load of 1 kg. By applying the proposed hybrid PID-ILC scheme and repeating its operation several times, the system stabilized further, and the steady-state error was reduced. Stability was reached under no-load conditions after 15 iterations, with an error of 0.319°. Under load, 20 iterations reduced the error to 0.305°. The controller remained stable throughout all tests and conditions, and errors continued to decrease with each new cycle. There was no disturbance, and an oscillation was present, ensuring that the response of the system was bounded. These findings demonstrate that the proposed method is a practical way to improve the precision, robustness, and reliability of inexpensive robotic arms in educational institutions or SMEs.

In summary, an experimental analysis showed that PID-ILC significantly improves the accuracy, stability, and robustness of a low-cost articulated robotic arm (SCORBOT ER4u) performing complex 3D trajectories. In the experiments, the steady-state error was reduced by 35% under no-load conditions, and by 71% with the 1 kg payload application, signifying the superiority of the proposed method that works well under nonlinear cases with disturbances. Therefore, this study is the very first experimental validation that these lightweight arms have improved tracking accuracy in the presence of disturbances from the payload, a contribution worthwhile for educational institutions and SMEs, where affordability and reliability stand side by side. Compared to previous work, the hybrid PID-ILC control has better steady-state accuracy and performance in nonlinear conditions with disturbances and loading variations. Whereas most previous studies considered only 2D trajectories or free load conditions, this approach was validated for 3D trajectories in the presence of external disturbance, therefore showing applicability both to educational and cost-effective industrial systems.

Although a simulated 3D trajectory and real hardware under load were utilized to test the proposed control method, it still needs to be validated in real cases, such as pick and place. Therefore, future research will investigate the application of the proposed control method to real activities to evaluate its resilience and suitability for industrial and educational robotic applications. To improve scalability in large or dynamic industrial environments, real-time adaptive tuning and disturbance compensation will also be investigated.

ACKNOWLEDGMENT

The authors are grateful to the Sanata Dharma Foundation in Yogyakarta, Indonesia, for funding their research and the Rajamangala University of Technology Thanyaburi (RMUTT), Thailand, for providing a facility for their research.

REFERENCES

[1] L. Monostori, "Cyber-physical Production Systems: Roots, Expectations and R&D Challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014, https://doi.org/10.1016/j.procir.2014.03.115.

[2] K. Baizid *et al.*, "IRoSim: Industrial Robotics Simulation Design Planning and Optimization platform based on CAD and knowledgeware technologies," *Robotics and Computer-Integrated Manufacturing*, vol. 42, pp. 121–134, Dec. 2016, https://doi.org/10.1016/j.rcim.2016.06.003.

[3] R. U. Islam, J. Iqbal, S. Manzoor, A. Khalid, and S. Khan, "An autonomous image-guided robotic system simulating industrial applications," in *2012 7th International Conference on System of Systems Engineering (SoSE)*, Genova, Italy, Jul. 2012, pp. 344–349, https://doi.org/10.1109/SYSoSE.2012.6384195.

[4] Y. Kim and S. Kim, "Automation and Optimization of Food Process Using CNN and Six-Axis Robotic Arm," *Foods*, vol. 13, no. 23, Nov. 2024, Art. no. 3826, https://doi.org/10.3390/foods13233826.

[5] F. Cepolina and R. Razzoli, "Review of robotic surgery platforms and end effectors," *Journal of Robotic Surgery*, vol. 18, no. 1, Feb. 2024, Art. no. 74, https://doi.org/10.1007/s11701-023-01781-x.

[6] C. Urrea and J. Kern, "Recent Advances and Challenges in Industrial Robotics: A Systematic Review of Technological Trends and Emerging Applications," *Processes*, vol. 13, no. 3, Mar. 2025, Art. no. 832, https://doi.org/10.3390/pr13030832.

[7] V. Román-Ibáñez, F. Pujol-López, H. Mora-Mora, M. Pertegal-Felices, and A. Jimeno-Morenilla, "A Low-Cost Immersive Virtual Reality System for Teaching Robotic Manipulators Programming," *Sustainability*, vol. 10, no. 4, Apr. 2018, Art. no. 1102, https://doi.org/10.3390/su10041102.

[8] E. Ananias and P. D. Gaspar, "A Low-Cost Collaborative Robot for Science and Education Purposes to Foster the Industry 4.0 Implementation," *Applied System Innovation*, vol. 5, no. 4, Jul. 2022, Art. no. 72, https://doi.org/10.3390/asi5040072.

[9] M. Tiboni *et al.*, "Low-Cost Design Solutions for Educational Robots," *Journal of Robotics and Mechatronics*, vol. 30, no. 5, pp. 827–834, Oct. 2018, https://doi.org/10.20965/jrm.2018.p0827.

[10] H. M. Ali, Y. Hashim, and G. A. Al-Sakkal, "Design and implementation of Arduino based robotic arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, Apr. 2022, Art. no. 1411, https://doi.org/10.11591/ijece.v12i2.pp1411-1418.

[11] C. Zeng, H. Zhou, W. Ye, and X. Gu, "iArm: Design an Educational Robotic Arm Kit for Inspiring Students' Computational Thinking," *Sensors*, vol. 22, no. 8, Apr. 2022, Art. no. 2957, https://doi.org/10.3390/s22082957.

[12] K. Chenchireddy, R. Dora, G. B. Mulla, V. Jegathesan, and S. A. Sydu, "Development of robotic arm control using Arduino controller," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, no. 3, Sep. 2024, Art. no. 264, https://doi.org/10.11591/ijra.v13i3.pp264-271.

[13] Y. H. T. Htun, M. S. Hlaing, and T. T. Hla, "Master-Slave Synchronization of Robotic Arm using PID Controller," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 11, no. 1, pp. 77–87, Jan. 2023, https://doi.org/10.52549/ijeei.v11i1.4171.

[14] P. Sutyasadi, M. B. Wicaksono, and D. Maneetham, "Improvement Control of a Three Axis Articulated Robotic Arm Using PID Cascade Control," in *2023 11th International Conference on Cyber and IT Service Management (CITSM)*, Makassar, Indonesia, Nov. 2023, pp. 1–4, https://doi.org/10.1109/CITSM60085.2023.10455548.

[15] K. Ali, S. Ullah, A. Mehmood, H. Mostafa, M. Marey, and J. Iqbal, "Adaptive FIT-SMC Approach for an Anthropomorphic Manipulator With Robust Exact Differentiator and Neural Network-Based Friction Compensation," *IEEE Access*, vol. 10, pp. 3378–3389, 2022, https://doi.org/10.1109/ACCESS.2021.3139041.

[16] P. Chotikunnan and R. Chotikunnan, "Dual Design PID Controller for Robotic Manipulator Application," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, pp. 23–34, Feb. 2023, https://doi.org/10.18196/jrc.v4i1.16990.

[17] A. N. Wazzan, N. Basil, M. Raad, and H. K. Mohammed, "PID Controller with Robotic Arm using Optimization Algorithm," *International Journal of Mechanical Engineering*, vol. 7, no. 2, pp. 3746–3751, 2022.

[18] S. Müftü and B. Gökçe, "Design and Implementation of an Optimized PID Controller for Two-Limb Robot Arm Control," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 13, no. 1, pp. 192–204, Mar. 2024, https://doi.org/10.17798/bitlisfen.1370223.

[19] J. O. Pedro and R. V. Smith, "Real-Time Hybrid PID/ILC Control of Two-Link Flexible Manipulators," *IFAC-PapersOnLine*, vol. 50, no. 2, pp. 145–150, Dec. 2017, https://doi.org/10.1016/j.ifacol.2017.12.027.

[20] E. Parikesit, D. Maneetham, and P. Sutyasadi, "Control of robot-assisted gait trainer using hybrid proportional integral derivative and iterative learning control," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, Dec. 2022, Art. no. 5967, https://doi.org/10.11591/ijece.v12i6.pp5967-5978.

[21] J. Iqbal, N. G. Tsagarakis, and D. G. Caldwell, "A multi-DOF robotic exoskeleton interface for hand motion assistance," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, Aug. 2011, pp. 1575–1578, https://doi.org/10.1109/IEMBS.2011.6090458.

[22] J. Iqbal, O. Ahmad, and A. Malik, "HEXOSYS II - towards realization of light mass robotics for the hand," in *2011 IEEE 14th International Multitopic Conference*, Karachi, Pakistan, Dec. 2011, pp. 115–119, https://doi.org/10.1109/INMIC.2011.6151454.

[23] J. L. Lin, H. P. Huang, and C. Y. Lin, "Iterative Learning Control for Vibration Suppression of a Robotic Arm," *Applied Sciences*, vol. 13, no. 2, Jan. 2023, Art. no. 828, https://doi.org/10.3390/app13020828.

[24] O. Saleem, A. Hamza, and J. Iqbal, "A Fuzzy-Immune-Regulated Single-Neuron Proportional–Integral–Derivative Control System for Robust Trajectory Tracking in a Lawn-Mowing Robot," *Computers*, vol. 13, no. 11, Nov. 2024, Art. no. 301, https://doi.org/10.3390/computers13110301.

[25] O. Saleem, S. Ali, and J. Iqbal, "Robust MPPT Control of Stand-Alone Photovoltaic Systems via Adaptive Self-Adjusting Fractional Order PID Controller," *Energies*, vol. 16, no. 13, Jun. 2023, Art. no. 5039, https://doi.org/10.3390/en16135039.

[26] S. Astutik, Supeno, S. H. B. Prastowo, T. Prihandono, and S. Bektiarso, "Study of Kinematics and Dynamics of motion at Semanggi Bridge Jember, Indonesia as a contextual in Physics Learning," *Journal of Physics: Conference Series*, vol. 1832, no. 1, Mar. 2021, Art. no. 012033, https://doi.org/10.1088/1742-6596/1832/1/012033.

[27] I. Agustian, N. Daratha, R. Faurina, A. Suandi, and S. Sulistyaningsih, "Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit Hartenberg," *Jurnal Elektronika dan Telekomunikasi*, vol. 21, no. 1, Aug. 2021, Art. no. 8, https://doi.org/10.14203/jet.v21.8-18.

[28] J. A. Medrano-Hermosillo, A. E. Rodriguez-Mata, V. A. Gonzalez-Huitron, R. Sandoval, L. Djilali, and O. J. Suarez-Sierra, "Analysis of Forward Kinematics in Robotic Manipulators: Approaches Using Homogeneous Matrices, Screw Theory, Quaternions, and Geometric Algebra," *International Journal of Innovative Science, Engineering & Technology*, vol. 11, no. 11, Nov. 2024.

[29] Q. Niu, J. Zhao, L. Liang, J. Xing, H. Li, and Z. Wang, "A general framework for the analytical inverse kinematics solution of industrial robots," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 239, no. 12, pp. 4499–4511, Jun. 2025, https://doi.org/10.1177/09544062251318236.

[30] N. T. Dantam, "Robust and efficient forward, differential, and inverse kinematics using dual quaternions," *The International Journal of Robotics Research*, vol. 40, no. 10–11, pp. 1087–1105, Sep. 2021, https://doi.org/10.1177/0278364920931948.

[31] Q. Yang, F. Zhang, and C. Wang, "Deterministic Learning-Based Neural PID Control for Nonlinear Robotic Systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 5, pp. 1227–1238, May 2024, https://doi.org/10.1109/JAS.2024.124224.

[32] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, Jun. 2021, https://doi.org/10.1007/s40435-020-00665-4.

[33] A. R. Al Tahtawi, F. S. Putri, and M. Martin, "Position control of AX-12 servo motor using proportional-integral-derivative controller with particle swarm optimization for robotic manipulator application," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 12, no. 2, Jun. 2023, Art. no. 184, https://doi.org/10.11591/ijra.v12i2.pp184-191.

[34] V. T. Ha, T. T. Cao, T. T. Than, T. T. Nguyen, and H. D. B. Thi, "Learning and Uncertainty Compensation in Robotic Motion Systems using Li-Slotine Adaptive Tracking and Intelligent Adaptive Control," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20461–20470, Feb. 2025, https://doi.org/10.48084/etasr.9843.

[35] M. Hofer, L. Spannagl, and R. D'Andrea, "Iterative Learning Control for Fast and Accurate Position Tracking with an Articulated Soft Robotic Arm," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019, pp. 6602–6607, https://doi.org/10.1109/IROS40897.2019.8967636.

[36] P. W. Oliveira, G. A. Barreto, and G. A. P. Thé, "A General Framework for Optimal Tuning of PID-like Controllers for Minimum Jerk Robotic Trajectories," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 3–4, pp. 467–486, Sep. 2020, https://doi.org/10.1007/s10846-019-01121-y.

[37] Scorbot Er4u User Manual. intelitek, 2001.

[38] V. Tinoco, M. F. Silva, F. N. Santos, R. Morais, S. A. Magalhães, and P. M. Oliveira, "A review of advanced controller methodologies for robotic manipulators," *International Journal of Dynamics and Control*, vol. 13, no. 1, Jan. 2025, Art. no. 36, https://doi.org/10.1007/s40435-024-01533-1.

[39] C. Zhao and S. Yuan, "Tracking performance of PID for nonlinear stochastic systems." arXiv, 2023, https://doi.org/10.48550/ARXIV.2303.10537.

[40] M. Ali and H. Mirinejad, "Robust tracking control of flexible manipulators using hybrid backstepping/nonlinear reduced-order active disturbance rejection control," *ISA Transactions*, vol. 149, pp. 229–236, Jun. 2024, https://doi.org/10.1016/j.isatra.2024.04.026.