IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 30, 2019, accepted August 23, 2019, date of publication September 2, 2019, date of current version September 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939050

Intelligent Control Strategy for Robotic Arm by Using Adaptive Inertia Weight and Acceleration Coefficients Particle Swarm Optimization

TZUU-HSENG S. LI[1], (Member, IEEE), PING-HUAN KUO[2], YA-FANG HO[1], AND GUAN-HONG LIOU[1]

[1]aiRobots Laboratory, Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan
[2]Computer and Intelligent Robot Program for Bachelor Degree, National Pingtung University, Pingtung 90004, Taiwan

Corresponding author: Ping-Huan Kuo (phkuo@mail.nptu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2218-E-153-001-MY3 and Grant MOST 106-2221-E-006-009-MY3.

ABSTRACT This paper proposes an intelligent control strategy for enabling a robotic arm to grasp and place water-filled bottles without spilling any of the water. First, the system architecture of a five-degree-of-freedom robotic arm and its mechanical design are introduced. Second, both the forward and inverse kinematics of the robotic arm are derived. The study conducted an experiment in which the designed and implemented robotic arm could grasp a bottle of water and move it to another place. However, if the acceleration or the orientation of the robotic arm were inappropriate, the water in the bottle may be spilled during the movement. Therefore, the proposed strategy applies an inertial measurement unit for obtaining relevant information. According to the obtained information, the velocity curves of each joint could be optimized by adaptive inertia weight and acceleration coefficients particle swarm optimization. Finally, the experimental results demonstrated the feasibility and effectiveness of the proposed method.

INDEX TERMS AIWCPSO, intelligent control, robotic arm, velocity control.

I. INTRODUCTION

In recent years, industrial automation has attracted attention worldwide and has been actively promoted. Robots play an essential role in industrial automation. Software and hardware related to robots have been upgraded constantly, and robots with incrementally superior capacities have been invented. Robots can assist humans in tasks or conduct hazardous tasks for humans. Therefore, robots are also crucial for humans.

Industrial robots primarily exist in the form of robotic arms. In the early stage of commercial robotics, robotic arms were used in various industrial manufacturing tasks, and currently, their application scopes have expanded. From the deployment of the first robotic arm to current industrial robots, related technology has improved and still continues to improve. Academics have published numerous studies on robotic arms. Various technology companies devote effort and resources to researching and developing robotic arms [1]–[5]. With advancements in software and hardware technology, computer processors have been miniaturized, but their performance levels have increased substantially. Over the past few years, the computation speed of processors has doubled. Additionally, motor technology and performance have advanced. Advancements in various fields have necessitated the expansion of robotic arm applications. The capability of robotic arms has been improved. Initially, robotic arms could only move objects. Currently, they can conduct more difficult tasks such as welding.

Different situations have different demands for robotic arms. To date, robotic arms have been applied in various fields. The researchers in [6] proposed a high-speed and lightweight robotic arm for playing badminton. In [7], a two-degree-of-freedom (DOF) fabric-based soft robotic arm was developed for grasping objects. Moreover, the authors in [8] produced a robotic arm for edge-rolling manipulation by proposing a grasp planning approach. In [9], a control method that enables a mechanical arm to flip burgers was proposed. The researchers in [10] utilized shape memory alloy coils to produce a soft robotic arm. A study [11] presented an

The associate editor coordinating the review of this manuscript and approving it for publication was Jinguo Liu.

innovative, humorous humanoid robot that could produce quick-and-wide arm motions.

Research has also investigated robotic arm control. For example, the researchers in [12] used H_∞ control design for motion control in elastic-joint robots and yielded favorable results. In [13], an adaptive impedance control method based on the concept of reinforcement learning was proposed; the proposed controller was applied to human–robot cooperation. The results demonstrated the effectiveness of the proposed controller in for performing cooperative human–robot tasks. The researchers in [14] proposed a multi-objective optimization method for position control of an arm module; the proposed method was effective and contributed to the development of an assistive device. Another study [15] proposed the application of a physical–virtual impedance control system to a dual-arm aerial manipulator; the study achieved favorable results. Research has also been conducted on collision estimation for robots working with humans. For example, a study [16] proposed a method that involves the projection of a human arm motion into the robot's path as a preventive measure; this contributed to future human–machine cooperation. To resolve various problems such as collisions, joint limits, and singularities, another study [17] proposed haptic guidance methods for a dual-arm system that achieved favorable verification. In [18], a stable control framework was proposed and applied to a lightweight dual-arm robot. The experimental results showed that this framework could achieve favorable results even under conditions of high uncertainty.

During the use of a robotic arm to grasp objects, velocity control and endpoint angle settings are critical. In the present study, for example, when a robotic arm was used to move a liquid-filled bottle in 3 s, the liquid spilled. The spillage can be ascribed to two main reasons. First, during the movement process, the gripper was not horizontal; it was tilted slightly. Second, the velocity allocation during the movement process was inadequate. In the process of moving an object in a short period, maximum acceleration tends to be excessively large, causing the surface of the water to shake excessively and thus resulting in water spillage.

To resolve this problem, this study proposes a smart control strategy for a robotic arm. The strategy involves a relatively modest learning process that leads to substantial improvement. A robotic arm with a total length and weight of 66.01 cm and 2.7 kg, respectively, was applied in this study. Iterative design and improvement produced a rigid robotic arm whose weight could be sustained by the available motors.

Particle swarm optimization (PSO) has an extensive application range [19]–[26]. This method has been expanded to various forms, such as linear time-varying PSO [27], random PSO [28], rank-based PSO [29], increasing PSO [30], adaptive inertia weight PSO (AIWPSO) [31], and AIWCPSO [32], where AIWCPSO was adopted from AIWPSO. Studies have argued that AIWCPSO has superior learning effectiveness. Accordingly, this study used the AIWCPSO method.

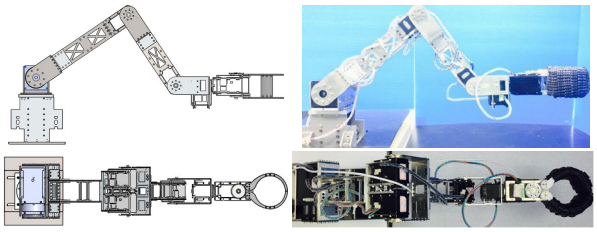


FIGURE 1. Design of robotic arm mechanism.

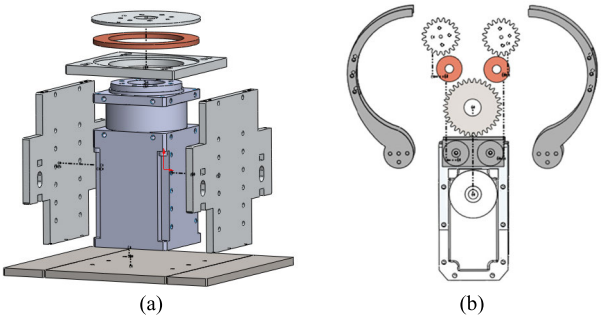
The smart control strategy was designed using AIWCPSO [32] and velocity curves. An inertial measurement unit (IMU) sensor located at the end of the machine arm transmit the status of the gripper during movement to the smart control system for calculation and learning. After multiple iterations of a learning loop, the motor at each joint can obtain an optimal velocity curve; thus, during the entire movement process, liquid spillage would not occur.

The major contributions of this study are outlined as follows: 1) applying the AIWCPSO algorithm to obtain an optimal velocity control curve; 2) defining the velocity control curve module for the robotic arm; 3) designing a robotic arm for the proposed intelligent control strategy; and 4) proving the practicality and feasibility of the proposed velocity control strategy for the designed robotic arm.

The rest of this paper is organized as follows. In Section II, the mechanisms of the robotic arm are discussed. The optimization algorithm used for the designed robotic arm is described in Section III. In Section IV, experimental results are presented to support the argument that the proposed control strategy is feasible and practical. Finally, Section V concludes this paper.

II. MECHANISMS OF THE ROBOTIC ARM

Fig. 1 illustrates the design of the robotic arm. The robotic arm has five DOF. It contains seven dc servomotors. The total weight and length of the arm were measured to be 2.7 kg and 66.01 cm, respectively. The robot's metal components are composed of three metal types: 6061 aluminum alloy, 7075 aluminum alloy, and stainless steel. Among the three metals, 6061 aluminum alloy has the lowest density, followed by 7075 aluminum alloy and then stainless steel. Because sheet metal components require bending, high-density materials are not suitable. Therefore, most of the sheet metal components are composed of 6061 aluminum alloy. By contrast, the metal components at the robotic arm's base that does not require bending are all composed of 7075 aluminum alloy, which is approximately as dense as stainless steel; therefore, it is both rigid and lighter relative to stainless steel of the same volume. Stainless steel constitutes the metal components near the robotic arm's base that does not require bending. Its high density results in excellent rigidity, and its weight is three times heavier than 7075 aluminum alloy of the same volume. Because stainless steel constitutes the metal components at the base, its weight rests directly on the desk and does not impose excessive burdens on the motors.



(a) (b)

FIGURE 2. Base and gripper design: (a) base and (b) gripper.

To reduce the total weight of the robotic arm, the gripper and the gears are made of glass fibers and reinforced plastics.

Fig. 2 presents the design of the base and the gripper. The base must fix the arm-connected extrusion to the desktop. The first motor must support the weight of the entire robotic arm and must frequently rotate at high speed; hence, it is equipped with a thrust-bearing needle roller. When the motor rotates, the push force induces the needle to roll upward, minimizing friction and thus minimizing the loading of the motor.

The motor to the gripper uses a 32-tooth gear, whereas the gripper uses a smaller 22-tooth gear. Therefore, the dc servomotor is required to rotate by only 62°, and the gripper can change from an open position to a closed position to grasp objects. The gripper contains a thrust-bearing needle roller to sustain the weight of the held object. When the gripper is loaded, the friction caused by opening and closing is minimized. The bearing is in the robotic arm.

Regarding the robotic arm mechanism, the dc servomotors connect metal components and have metal support designs to increase the rigidity of the arm joints in order to enhance stability.

In the system framework, a notebook computer serves as the central processing system and is responsible for smart control strategy learning and computing, kinetic computing, velocity curve design for the motors at each joint, acceleration sensing, and gyro information processing. Information detected by the sensors is transmitted to the central processing system in its entity. The smart control strategy applies this information to execute the learning process.

Regarding the robotic arm control, the most crucial part is the communication between the robotic arm and the central processing system; specifically, a communication system must be designed to precisely transmit orders, based on data calculated by the central processing system, to the robotic arm so that it can precisely implement ordered actions (Fig. 3). Accordingly, in the proposed strategy, USB2Dynamixel is used to transmit data between the robotic arm and the central processing system. A half-duplex transmission mode is applied, signifying that the two devices (notebook and arm) can conduct two-way data transmission. Moreover, only one device can transmit data at one time, and the other one must be in receiving mode. Therefore, during the control of robotic arms, if a motor is instructed to transmit its current state, the data transmission cycle would be long.

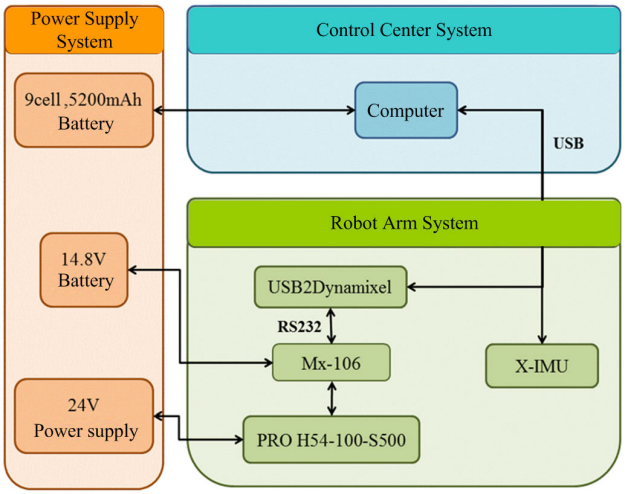


FIGURE 3. Schematic of robotic arm hardware.

The connection between the acceleration sensor, gyro, and central processing system is also critical. The learning process requires the sensor to transmit data. Therefore, continuous information transmission is established to accurately obtain the motion of the robotic arm at each time point. These data enable the smart control strategy to continue to learn about the accurate direction, finally presenting favorable results.

III. OPTIMIZATION ALGORITHM

Eberhart and Kennedy developed the PSO algorithm inspired by the social behavior of bird flocking or fish schooling [33], [34]. These animals utilize their own experiences and refer to others' experiences to determine their direction. Therefore, the PSO algorithm has the advantages of rapid convergence and simple concept, and it has been widely adopted. The PSO has provided favorable solutions to numerous problems. However, this algorithm still has weaknesses. For example, it easily leads to local optimal solutions. Scholars have attempted to improve the PSO algorithm. Nichabadi *et al*. proposed the AIWPSO algorithm [31] that changes the original fixed inertia weights to dynamic weights, in addition to adjusting the particle learning situation. The adaptive inertia weights are adjusted according to the updated particles of the previous generation. The acceleration parameter is still a constant. This algorithm reduces the tendency of arriving at a local optimal solution and can lead to favorable learning results. The AIWCPSO algorithm is an improvement that is primarily based on the AIWPSO algorithm.

Two types of the PSO algorithm are detailed in the following sections. The first type is the standard PSO, in which the inertia weight and acceleration parameters are all constant.

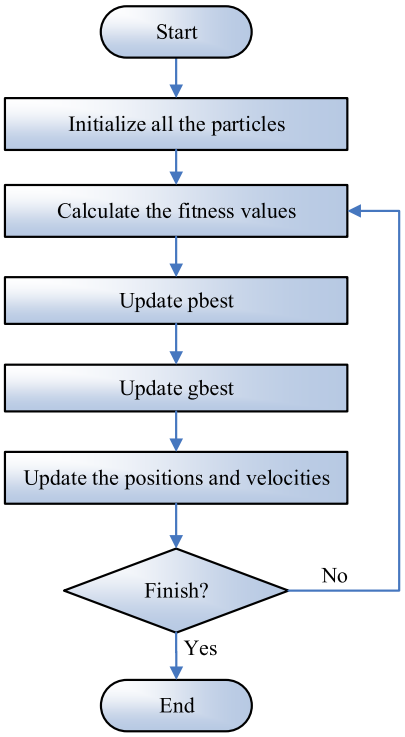


FIGURE 4. Schematic of PSO algorithm.

The second type is the AIWCPSO algorithm, which is different from the standard PSO algorithm.

A. STANDARD PSO

Fig. 4 illustrates a schematic of the standard PSO algorithm. The operating procedure of this algorithm is divided into five stages. The first stage, involves particle initialization. In this stage, the location and velocity of each particle would be assigned a random number. The second stage entails calculating a fitness function according to the needs of the question. Every time a particle updates its location and velocity, the fitness function determines the fitness of that particle due to its movement. The third stage involves updating the personal best of each updated particle. The personal best of each particle is recorded. When the fitness function exceeds the personal best, the personal best is updated. The fourth state entails updating the global best, which is the optimal solution of all particles. After the learning of each generation of particles, their personal best data are compared with the global best. If their data are superior to the global best, the global best is updated. The fifth stage involves updating the velocity and location of each particle. The velocity is influenced by the current velocity, personal best, and global best, as presented in (1). The location of a particle at the next time point is determined by the current location plus the velocity of the particle, as indicated in (2). The fifth stage involves establishing a program termination condition. Different problems lead to different termination conditions, such as whether an error must be smaller than a tolerable error and whether a certain number of iterations has been reached.

$$V(t+1) = w \times V(t) + c_1 \times rand_1 \times (pbest - X(t)) + c_2 \times rand_2 \times (gbest - X(t)) \quad (1)$$

$$X(t+1) = X(t) + V(t+1) \quad (2)$$

where w is the inertia weight and c_1 and c_2 are acceleration parameters. These three parameters are constant during the PSO learning period. Moreover, $rand_1$ and $rand_2$ are random numbers between 0 and 1; $pbest$ is the personal best, representing the optimal location of a particle; $gbest$ is the global best, derived from the optimal positions of all particles; $V(t)$ is the current velocity of the particle; $V(t+1)$ is the velocity of the particle at the next point in time; $X(t)$ is the location of the particle at time t; and $X(t+1)$ is the location of the particle at time $t+1$.

B. AIWCPSO ALGORITHM

Nickabadi *et al*. proposed the AIWPSO algorithm [31], which is an improvement of the PSO algorithm in that it transforms the constant inertia weight to a dynamic weight. The conditions in (3) are used for updating the personal best of each particle. If $S(i,t) = 1$, then the personal best of a particle is updated; the personal best is not updated if $S(i,t) = 0$. In (3), i is the particle number and t is the current iteration number. $P_s(t)$ is the ratio of the total personal best of a current particle against the total number of particles; n is the total number of particles, as shown in (4). Equation (5) presents the principle for updating the inertial weight, where $w_{\min}$ and $w_{\max}$ are the self-determined minimum and maximum values of the weight, respectively, and range between 0 and 1. Equation (1) can be modified into (6):

$$S(i,t) = \begin{cases} 1, & if\ fitness\left(pbest_i^t\right) > fitness\left(pbest_i^{t-1}\right) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$P_s(t) = \frac{\sum_{i=1}^{n} S(i,t)}{n} \quad (4)$$

$$w_a(t) = (w_{\max} - w_{\min}) P_s(t) + w_{\min} \quad (5)$$

$$V(t+1) = w_a(t) \times V(t) + c_1 \times rand_1 \times (pbest - X(t)) + c_2 \times rand_2 \times (gbest - X(t)) \quad (6)$$

Fig. 5 shows the flowchart of the AIWCPSO algorithm. The AIWCPSO algorithm is based on the main structure of the AIWPSO algorithm; thus, it maintains the advantages of the AIWPSO algorithm, apart from modifying its acceleration parameters [27]. The difference between the AIWCPSO and PSO algorithms is that the AIWCPSO algorithm updates the global best as well as the location and velocity of each particle. The AIWPSO algorithm also updates the adaptive inertia weight and adaptive acceleration parameters, thus transforming c_1 and c_2 in the PSO algorithm through $P_s(t)$ into the dynamic adaptive acceleration parameters $a_1(t)$ and $a_2(t)$, as presented in (7). α and β are positive values determined by users. Equation (6) can be modified to yield (8).

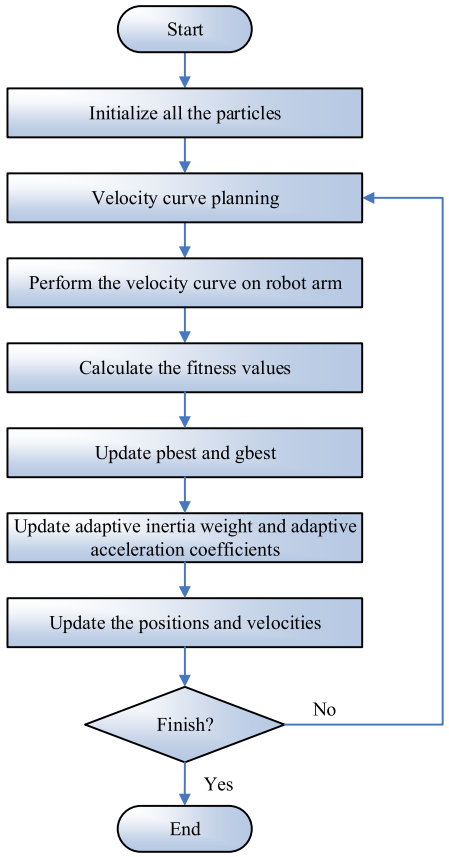


FIGURE 5. Flowchart of AIWCPSO algorithm.

Existing studies have shown that the AIWCPSO algorithm exhibited superior learning effects to the AIWPSO algorithm.

$$\begin{cases} a_1(t) = \alpha(P_s(t)) + \beta \\ a_2(t) = \alpha(1 - P_s(t)) + \beta \end{cases} \tag{7}$$

$$V(t+1) = w_a(t) \times V(t) + a_1(t) \times rand_1 \times (pbest - X(t)) + a_2(t) \times rand_2 \times (gbest - X(t)) \tag{8}$$

C. VELOCITY CURVE DESIGN

One of the crucial components of smart control strategies is velocity curve design. Fig. 6 presents a velocity curve, where the x-axis represents normalized time. A robotic arm operated in accordance with this velocity curve accelerates from a static state to the point of maximum angular velocity and then gradually slows down to the static state. A velocity curve is mainly composed of an acceleration terminal point t_1, deceleration starting point t_2, and maximum angular velocity. The velocity curve of each motor differs and has different t_1, t_2, and $\omega_{\max}$ values. A velocity curve can be divided into three stages, each of which is described in (9), where θ_1 in the first stage is shown in (10) and θ_2 in the third stage is shown in (11).

$$\begin{cases} \omega(t) = \dfrac{\omega_{\max}}{2\pi}(\theta_1 - \sin\theta_1), & 0 < t \le t_1 \\ \omega(t) = \omega_{\max}, & t_1 < t \le t_2 \\ \omega(t) = \dfrac{\omega_{\max}}{2\pi}(\theta_2 - \sin\theta_2), & t_2 < t \le 1 \end{cases} \tag{9}$$

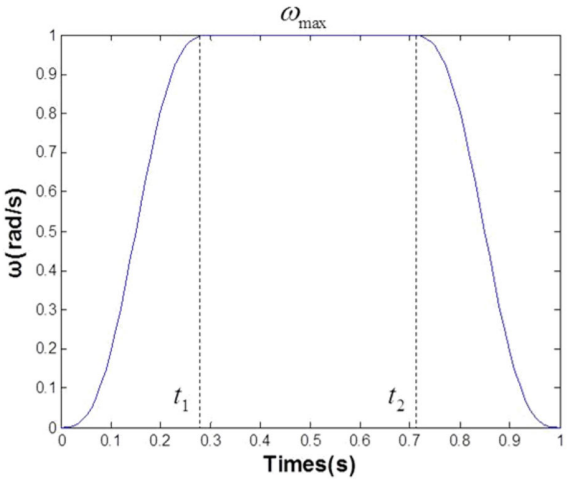


FIGURE 6. Velocity curves.

$$\theta_1 = \frac{t}{t_1}2\pi \tag{10}$$

$$\theta_2 = \frac{1-t}{1-t_2}2\pi \tag{11}$$

The velocity curve equation can be integrated to yield an equation of angular displacement $\Delta\theta$, where $\Delta\theta$, t_1, and t_2 are all known. Thus, the equation of maximum angular velocity $\omega_{\max}$ can be derived. The equations are presented in (12)–(16).

$$\Delta\theta = \int_0^{t_1} \frac{\omega_{\max}}{2\pi}\left(\frac{t}{t_1}2\pi - \sin\left(\frac{t}{t_1}2\pi\right)\right)dt + \int_{t_1}^{t_2} \omega_{\max}dt + \int_{t_2}^{1} \frac{\omega_{\max}}{2\pi}\left(\frac{1-t}{1-t_2}2\pi - \sin\left(\frac{1-t}{1-t_2}2\pi\right)\right)dt \tag{12}$$

$$\Delta\theta = \left(\frac{\omega_{\max} \times t^2}{2 \times t_1} + \frac{\omega_{\max} \times t_1 \times \cos\left(\frac{2\times\pi\times t}{t_1}\right)}{4 \times \pi^2}\right)\Bigg|_0^{t_1} + (\omega_{\max} \times t)|_{t_1}^{t_2} + \left(\frac{\omega_{\max} \times (1-t)^2}{2 \times (1-t_2)} + \frac{\omega_{\max} \times (1-t_2) \times \cos\left(\frac{2\times\pi\times(1-t)}{(1-t_2)}\right)}{4 \times \pi^2}\right)\Bigg|_{t_2}^{1} \tag{13}$$

$$\Delta\theta = \frac{\omega_{\max} \times t_1}{2} + \omega_{\max} \times (t_2 - t_1) + \frac{\omega_{\max} \times (1-t_2)}{2} \tag{14}$$

$$\Delta\theta = \frac{\omega_{\max}(1 + t_2 - t_1)}{2} \tag{15}$$

$$\omega_{\max} = \frac{(2 \times \Delta\theta)}{(1 + t2 - t1)} \tag{16}$$

D. AIWPSO PARAMETER SETTING

One of the objectives of this study was to achieve favorable learning results though few iterations; therefore, the AIWCPSO algorithm was adopted. The number of iterations was set to 9, and the total number of particles was set to 7. The information of each particle included the acceleration end time and deceleration starting time of five motors.

The smart control strategy applies the AIWCPSO algorithm, with OPS serving as the main part with partial adjustment. The parameter settings in [32] were a $w_{\min}$ value of 0 and $w_{\max}$ value of 1. Therefore, the adaptive inertia weight can be calculated using (17). A large inertia weight is conducive to ensure that a particle searches an entire region, whereas a small inertia weight limits the scope of the regional search process. Regarding the derivation of the acceleration parameters, α was set to 2.05 and β was set to 0.5; accordingly, the acceleration parameters could be derived as expressed in (18)

$$w_a(t) = P_s(t) \tag{17}$$

$$\begin{cases} a_1 = 2.05 \times (P_s(t)) + 0.5 \\ a_2 = 2.05 \times (1 - P_s(t)) + 0.5 \end{cases} \tag{18}$$

The goal of this study was to prevent shakiness due to acceleration and tilted grippers during the movement of the robotic arm, which would lead to spilling. Therefore, in our adaptive function design, we must consider four criteria: the peak values of the x-, y-, and z-axis accelerations as well as a summary of the differences between the gripper and the horizontal gesture at each time point, as shown in (19), as shown at the bottom of this page. The accelerations in all axes must be normalized to avoid the creation of an excessively small acceleration that would cause an extremely high adaptive value. A small acceleration peak means less shaking. Furthermore, if the gripper of the robotic arm could be consistently horizontal during task execution, liquid spillage due to the tilted gripper can be prevented.

IV. EXPERIMENTAL RESULTS

Fig. 7 shows the experimental setup of this study. The start points of the three experimental routes in this study were the No. 1–3 locations on the brown platform, whereas the end points were the No. 4–6 locations on the red platform. During the learning process in the smart control strategy, movements were repeatedly conducted and a considerable amount of water could be spilled due to the tilting of the robotic arm or inappropriate velocity curve; therefore, translucent bottles with caps were used to replace cups to prevent spillage. After the completion of an experimental process, the cap of the water bottle was unscrewed to demonstrate the differences between learned and unlearned situations. Furthermore, due to spoilage during the practice process, all prompts were made of waterproof materials.

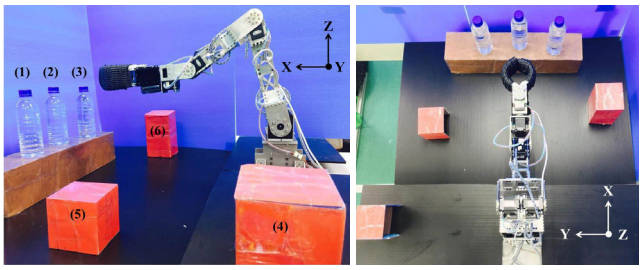


FIGURE 7. Experimental setup.

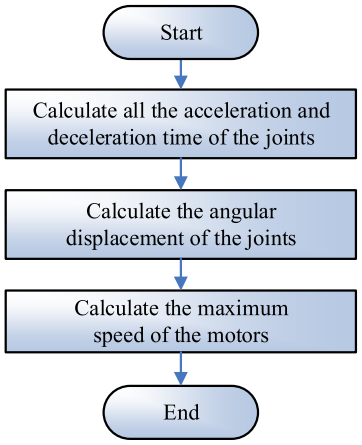


FIGURE 8. Flowchart of velocity design.

After defining the three routes, each of which was composed of locations with two coordinates, this study calculated the motor angle of each joint by using reverse kinetics. Fig. 7 shows the relative position of all objects. Route 1 involved taking up the water bottle from location No. 1 at (64.55, 14, 0) and moving it to location No. 4 at (4, 46.2, 7.2). Route 2 involved taking up the water bottle from location No. 2 at (66.05, 0, 0) and moving it to location No. 5 at (38.6, 32.4, −0.4). Finally, Route 3 involved taking up the water bottle from location No. 3 at (64.55, 14, 0) and moving it to location No. 6 at (41.9, −31.6, 7).

The gripper must be horizontal at the start and end points. The gripper could rotate about the z-axis only. The grasping gesture matrix is presented in (20), where θ could be obtained from the aforementioned six coordinates. The θ values at the start and end points of Route 1 were 12.24° and 85.05°, respectively; those at the start and end points of Route 2 were 0° and 40.01°, respectively; and those at the start and end

$$fitness = \frac{1}{\left|Acc_{x,\max}\right| + \left|Acc_{y,\max}\right| + \left|Acc_{z,\max}\right| + \left|\sum_{i=1}^{n} Gyro(i) - Gyro_{horizontal}\right|} \tag{19}$$

points of Route 3 were $-12.24°$ and $-37.02°$, respectively.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

A. EXPERIMENTAL PROCEDURES

The smart control strategy involves the AIWCPSO algorithm and velocity curve design. The operating procedure of the strategy involves five steps. The first step entails applying the AIWCPSO algorithm to update the acceleration stopping time and the deceleration starting time of each joint and particle. The second step involves designing a velocity curve for movement. The flowchart is illustrated in Fig. 8. Using reverse kinetics and the start and end points of each path, the system can obtain the initial and end angles of the motor of each joint, thereby obtaining the angular displacement of each motor. The point at the acceleration ending time and the deceleration starting time can be used to calculate the maximum angular velocity. The third step entails implementing the designed velocity curve on the robotic arm. The IMU sensor transmits the information of the gripper. The fourth step involves using the transmitted information to calculate fitness values according to four reference indicators: x-, y-, and z-axis maximum acceleration as well as the grasping gesture during the movement process. The final step entails updating the personal best, global best, adaptive inertia weight, and adaptive acceleration parameters of the particles. Steps 1–5 are repeated until the preset number of iterations was reached.

The proposed smart control strategy was verified by experiments in which a robotic arm grasped and placed a water-filled bottle by using three different routes. Regardless of distance, the robotic arm was required to complete the transfer within 3 s. Before the application of the smart control strategy, the robotic arm moved at an even speed. Without the smart strategy, at the beginning of the movement, the gripper tilted and the arm moved at a considerably high speed, causing the arm to hit the platform and spill the water. Additionally, when approaching its destination, the arm spilled the water again because of the large negative acceleration. These problems were corrected by the proposed smart control strategy.

Each route was provided with three types of information: first-generation particle and global best three-axis velocity curve graph, acceleration and gesture graph, and global best adaptive value graph. The x-axes of the three graphs represent time, with each unit representing 20 ms.

B. TRAINING ROUTE 1

Fig. 9 provides a breakdown of the movements along Route 1. Regarding the first-generation iteration, Fig. 10 illustrates the designed velocity curve for optimal particle velocity. The horizontal axis represents time, whereas the vertical axis represents the speed dial on the motor. The

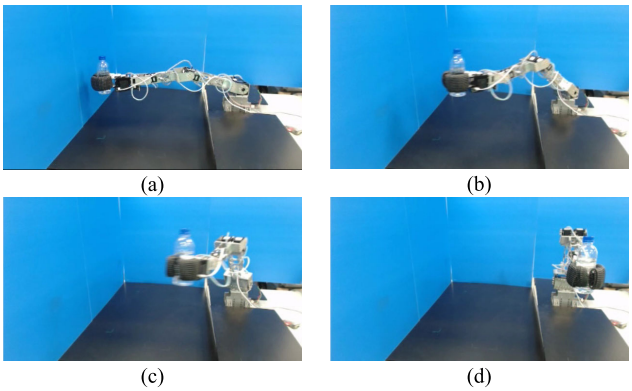


FIGURE 9. Motion breakdown along Route 1.

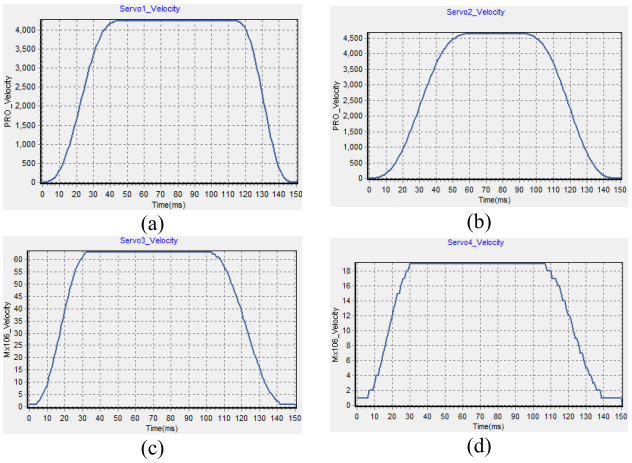


FIGURE 10. First-generation particle velocity design along Route 1. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

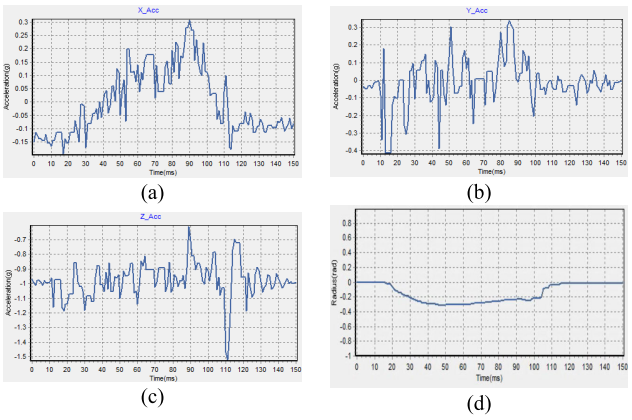


FIGURE 11. First-generation particle acceleration graph and gesture curve along Route 1. (a) x-axis acceleration curve, (b) y-axis acceleration curve, (c) z-axis acceleration curve, and (d) grasping gesture curve.

calculated velocity is transmitted to central processing system that commands the robotic arm to move accordingly.

Data transmitted from the IMU sensor on the gripper to the central processing system are graphed in Fig. 11. The three axial accelerations can be observed in Fig. 11. The x-axis represents time, and the y-axis represents acceleration. After the absolute value was taken, the peak accelerations in the x-,

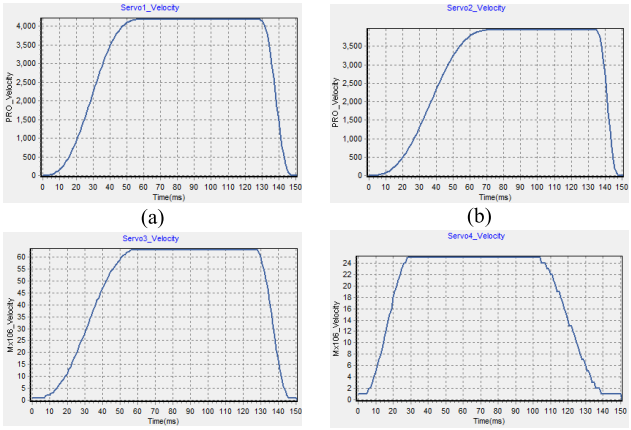


FIGURE 12. Global best velocity design for robotic arm along Route 1. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

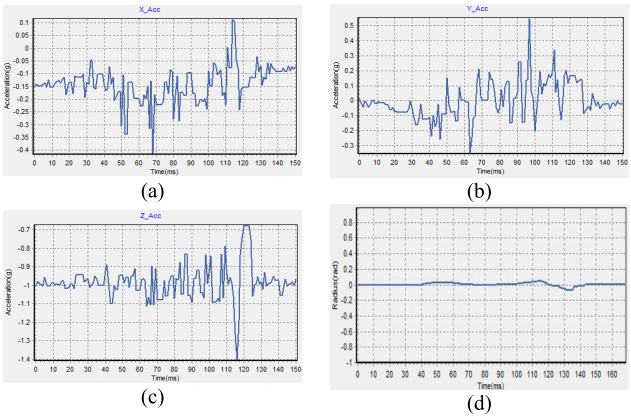


FIGURE 13. Global best acceleration curve and gesture curve along Route 1. (a) x-axis acceleration curve, (b) y-axis acceleration curve, (c) z-axis acceleration curve, and (d) grasping gesture curve.

y-, and z-axes were 0.32g, 0.42g, and 1.52g, respectively. The y-axis in Fig. 11(d) is demarcated in radians, with the 0 point signifying the gripper being horizontal. Therefore, during the movement of the robotic arm, the gripper was mostly tilted, leaving considerable room for improvement.

Through multiple sessions of learning, the global best along Route 1 was obtained. Subsequently, velocity curves were used to design the most suitable velocity curve (Fig. 12).

Fig. 13 illustrates data provided by the IMU sensor. The peak accelerations in the x- and y-axes were 0.41g and 0.32g, respectively, but the accelerations in both axes were almost steady. The peak acceleration in the z-axis was 1.4g. As indicated in Fig. 13(d), the grasping gestures throughout the movement process were almost horizontal. As illustrated in Fig. 14, when the AIWCPSO system was learning, nine iterations were required for determining the global best, and the adapted value of the global best was substantially improved subsequently. The learning time was 19.32 min. The grasping gesture reflected clear differences.

C. TRAINING ROUTE 2

Fig. 15 presents a breakdown of the movement of the robotic arm along Route 2. Fig. 16 presents the designed velocity

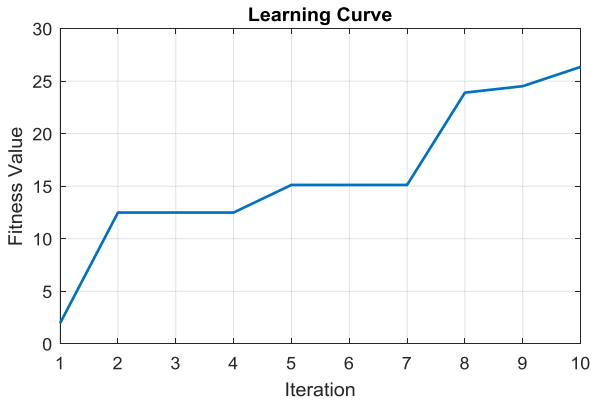


FIGURE 14. Learning curve for global best adaptive value along Route 1.

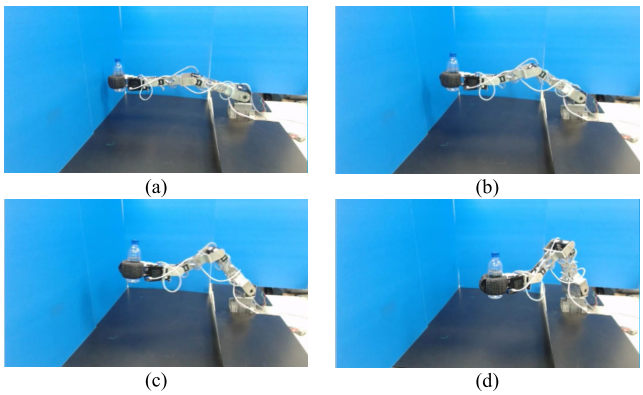


FIGURE 15. Movement along Route 2.

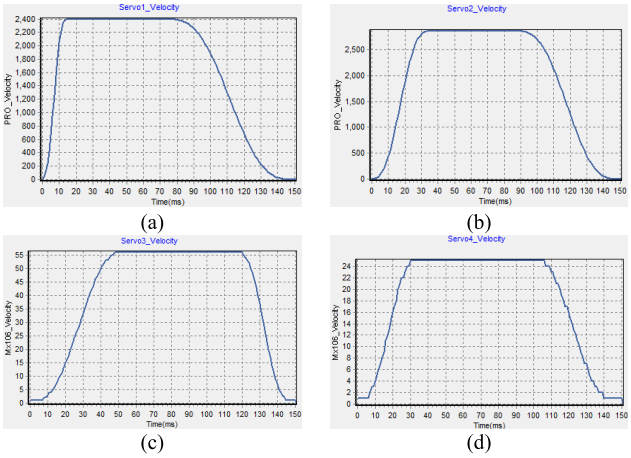


FIGURE 16. First-generation particle velocity designs along Route 2. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

curve for optimal particle velocity in the first-generation iteration. The horizontal axis represents time, whereas the vertical axis represents the speed dial on the motor.

Data transmitted from the IMU sensor on the gripper are shown in Fig. 17. The x-axis represents time, and the y-axis represents acceleration. The axial accelerations on the three axes are shown in the figure. The peak accelerations in the x-, y-, and z-axes were 0.31g, 0.41g, and 1.49g, respectively. As indicated in Fig. 17(d), the gripper tilted severely.

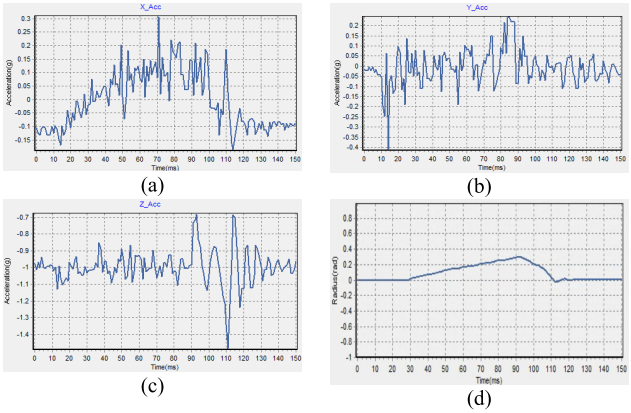


FIGURE 17. First-generation particle acceleration curves as well as gesture curve along Route 2. (a) *x*-axis acceleration curve, (b) *y*-axis acceleration curve, (c) *z*-axis acceleration curve, and (d) grasping gesture curve.

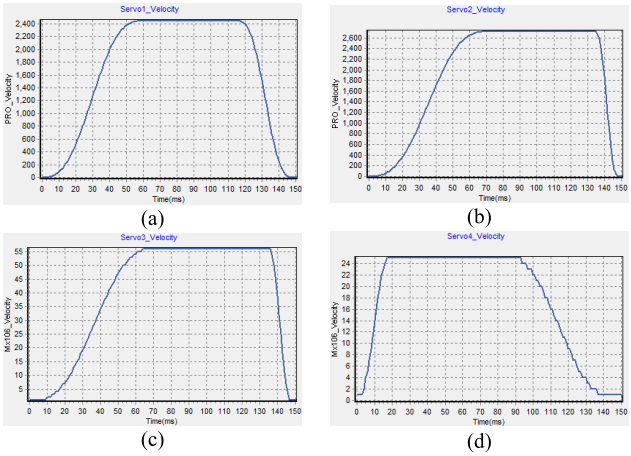


FIGURE 18. Global best velocity designs along Route 2. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

Fig. 18 illustrates the global best velocity curve along Route 2. Data from the IMU are shown in Fig. 19. The peak accelerations in the *x*- and *y*-axes were 0.27g and 0.28g, respectively, whereas the acceleration in the *z*-axis was mostly between 0.9g and 1.1g. The radian values in Fig. 19(d) are mostly 0, showing that the gripper maintained a horizontal orientation. Fig. 20 reveals the learning curve of the AIWCPSO algorithm for determining the global best. After a few iterations, the global best adaptive value improved by tenfold or more. The total learning time was 19.30 min. The gripper maintained a horizontal orientation, and the accelerations in the three axes approached stability. The control of peak acceleration in the *y*-axis was outstanding.

D. TRAINING ROUTE 3

Fig. 21 shows a breakdown of the robotic arm movements along Route 3. Fig. 22 presents the designed velocity curve for optimal particle velocity in the first-generation iteration. The horizontal axis represents time, whereas the vertical axis represents the speed dial on the motor.

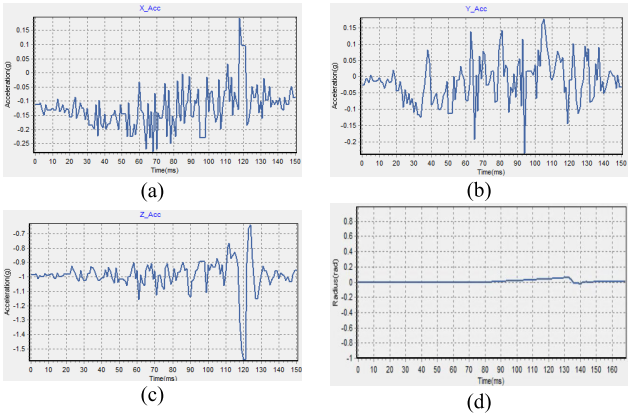


FIGURE 19. Global best acceleration curves and gesture curve along Route 2. (a) *x*-axis acceleration curve, (b) *y*-axis acceleration curve, (c) *z*-axis acceleration curve, and (d) grasping gesture curve.

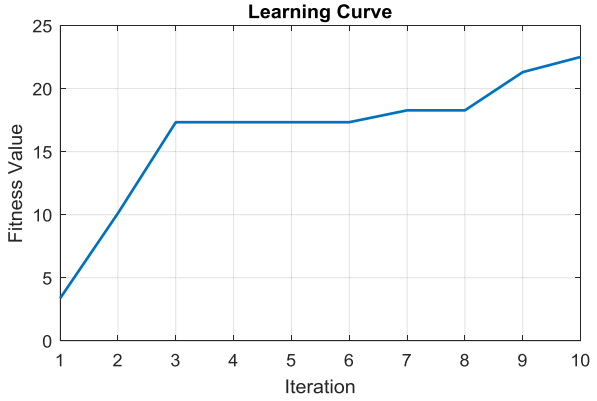


FIGURE 20. Learning curve for global best adaptive value along Route 2.

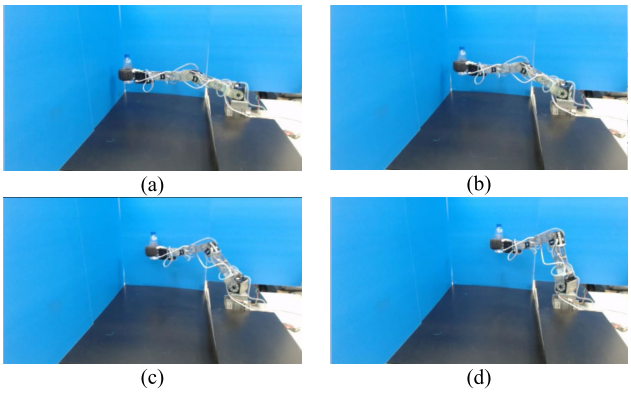


FIGURE 21. Movement along Route 3.

During the movement of the robotic arm, data were transmitted in real time from the IMU sensor. The transmitted data are depicted in Fig. 23. The *x*-axis represents time, and the *y*-axis represents acceleration. The accelerations in the three axes can be observed in the figure. The peak accelerations in the *x*-, *y*-, and *z*-axes were 0.25g, 0.17g, and 1.42g, respectively. As indicated in Fig. 23(d), the gripper tilted during the movement process.

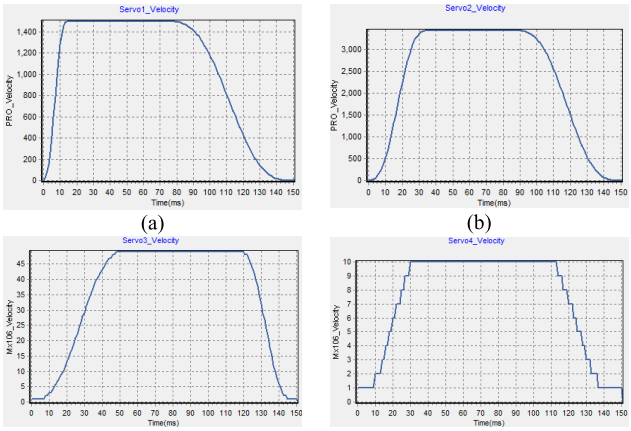


FIGURE 22. First-generation particle velocity designs along Route 3. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

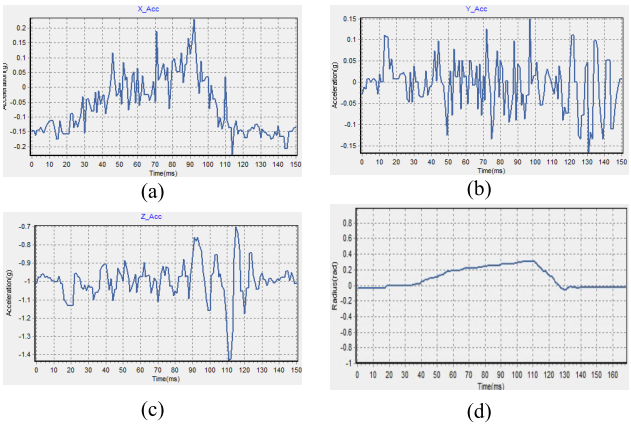


FIGURE 23. First-generation particle acceleration curves and gesture curve along Route 3. (a) *x*-axis acceleration curve, (b) *y*-axis acceleration curve, (c) *z*-axis acceleration curve, and (d) grasping gesture curve.

The global best along Route 3 was achieved by the third particle in the ninth generation. The system designed the velocity curve for optimal velocity along Route 3 (Fig. 24). Fig. 25 illustrates the IMU data, in which the accelerations in the three axes can be observed. The peak accelerations in the *x*-, *y*-, and *z*-axes were 0.27g, 0.17g, and 1.48g, respectively. Fig. 25(d) reveals that all radian values are almost 0 for the entire process, indicating that the gripper maintained a horizontal orientation. Fig. 26 presents the learning curve of the AIWCPSO algorithm for determining the global best. The global best adaptive value improved substantially. The learning time was 19.29 min. Along Route 3, the grasping gesture improved significantly.

E. APPLICATION OF GRASPING WATER-FILLED BOTTLES

The learning process of the smart control strategy was applied to a robotic arm grasping water-filled bottles and moving them on the three routes. Although each of the bottles was filled to a height of 1 mm from the top, signifying a nearly full bottle, water was not spilled during the movement process. Clearly, the smart control strategy yielded substantial

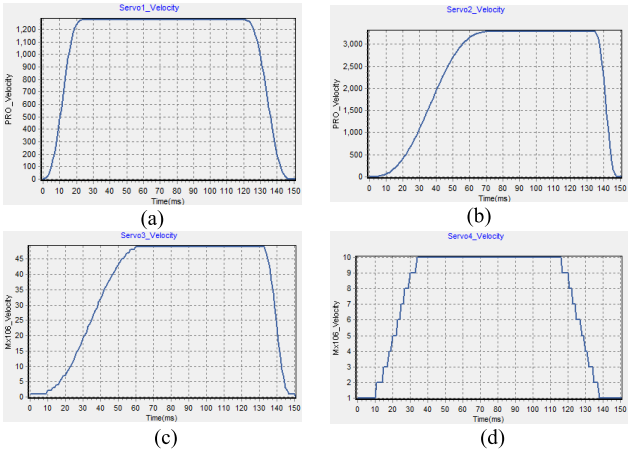


FIGURE 24. Global best velocity designs along Route 3. (a) Motor 1, (b) Motor 2, (c) Motor 3, and (d) Motor 4.

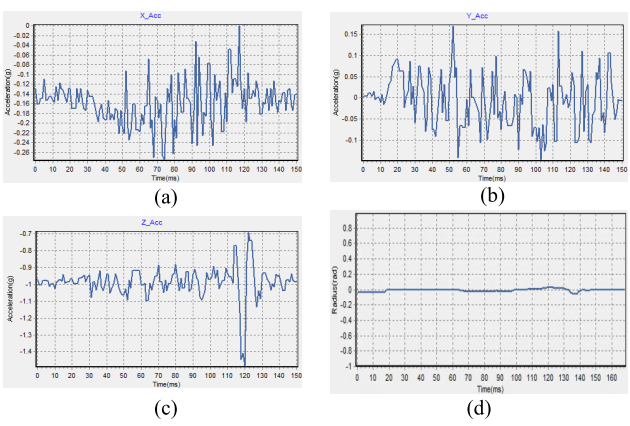


FIGURE 25. Global best acceleration curves and gesture curve along Route 3. (a) *x*-axis acceleration curve, (b) *y*-axis acceleration curve, (c) *z*-axis acceleration curve, and (d) grasping gesture curve.

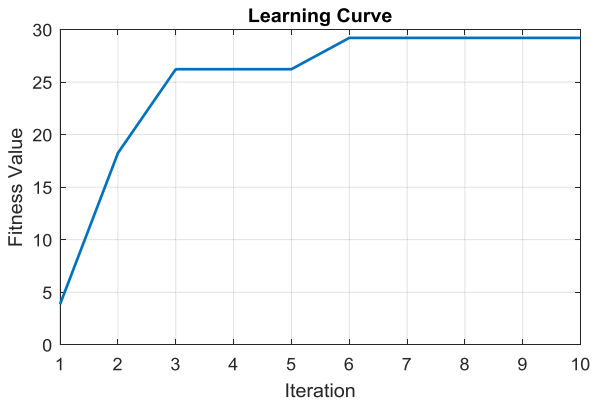


FIGURE 26. Learning curve for global best adaptive value along Route 3.

improvement because the gripper of the robotic arm maintained an excellent horizontal gesture. The system optimized inhibition of maximum acceleration and stabilized the three-axis acceleration.

The following figures depict the processes of grasping and moving the water-filled bottles along the three routes.

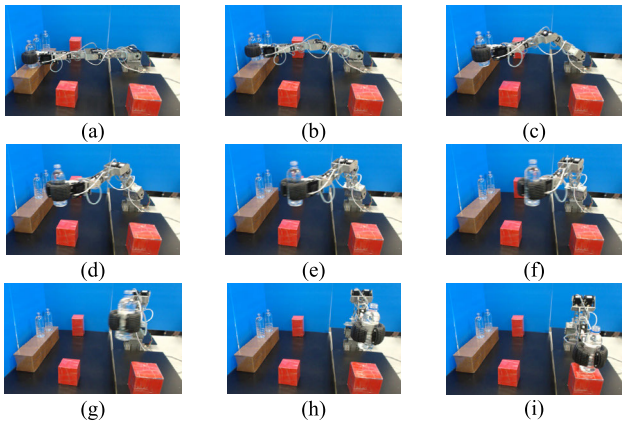


FIGURE 27. Grasping and movement of water-filled bottle along Route 1.

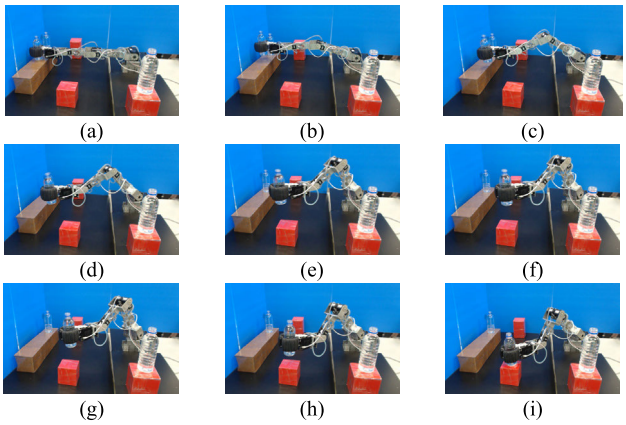


FIGURE 28. Gasping and movement of water-filled bottle along Route 2.

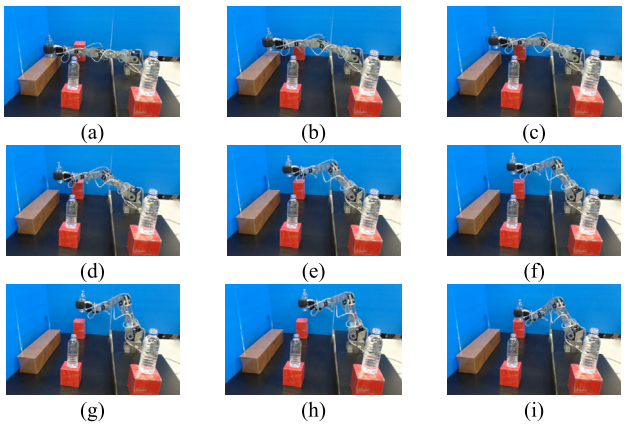


FIGURE 29. Gasping and movement of water-filled bottle along Route 3.

Figs. 27, 28, and 29 depict the movements along Routes 1, 2, and 3, respectively.

This section introduces the experimental environment, the flowchart of the smart control strategy, velocity curve design, AIWCPSO parameter settings, and grasping and movement of water-filled bottles. Regarding the grasping of the water-filled bottles, three learning routes are described in this section, along with information regarding acceleration, velocity curves, and grasping gesture. The application of results obtained from the learning process of the smart control strategy is also demonstrated.

After observing the x-, y-, and z-axis accelerations throughout the movement process, this study applied the proposed smart control strategy to reduce most of the excessively large accelerations, resulting in stabilized accelerations. The most obvious learning effect was in the robotic arm grasping gesture during the movement process. Before the learning process, the gripper was tilted throughout the process, but after the learning process, the gripper was almost always horizontal through the process. The whole experimental video can be accessed in [35].

V. CONCLUSION

This study has presented an intelligent control strategy and applied it to a self-made robotic arm to grasp and place water-filled bottles. The proposed intelligent control strategy adopted the AIWCPSO algorithm along with velocity curve design. The main goal of the experiments in this paper is to determine the optimal solution to overcome the problem of moving water-filled containers. After the learning process mentioned in the previous sections, the grasping gesture can be substantially improved; the gripper can also maintain a horizontal gesture during the movement process. Additionally, the accelerations in the x-, y-, and z-axes have been stabilized, and their peaks have been reduced. The results of this study demonstrate the feasibility and effectiveness of the proposed intelligent control strategy.

REFERENCES

[1] L. Minati, N. Yoshimura, and Y. Koike, "Hybrid control of a vision-guided robot arm by EOG, EMG, EEG biosignals and head movement acquired via a consumer-grade wearable device," *IEEE Access*, vol. 4, pp. 9528–9541, 2016.

[2] S. Li, J. Li, G. Tian, and H. Shang, "Stiffness adjustment for a single-link robot arm driven by series elastic actuator in muscle training," *IEEE Access*, vol. 7, pp. 65029–65039, 2019.

[3] W. Wang, C. Du, W. Wang, and Z. Du, "A PSO-optimized fuzzy reinforcement learning method for making the minimally invasive surgical arm cleverer," *IEEE Access*, vol. 7, pp. 48655–48670, 2019.

[4] B. Rouzbeh, G. M. Bone, G. Ashby, and E. Li, "Design, implementation and control of an improved hybrid pneumatic-electric actuator for robot arms," *IEEE Access*, vol. 7, pp. 14699–14713, 2019.

[5] Y. Zhang, W. Zhu, and A. Rosendo, "QR code-based self-calibration for a fault-tolerant industrial robot arm," *IEEE Access*, vol. 7, pp. 73349–73356, 2019.

[6] S. Mori, K. Tanaka, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "High-speed and lightweight humanoid robot arm for a skillful badminton robot," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1727–1734, Jul. 2018.

[7] X. Liang, H. Cheong, Y. Sun, J. Guo, C. K. Chui, and C.-H. Yeow, "Design, characterization, and implementation of a two-DOF fabric-based soft robotic arm," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2702–2709, Jul. 2018.

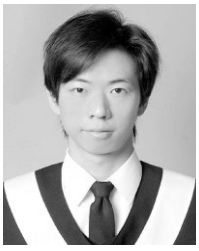
[8] A. Specian, C. Mucchiani, M. Yim, and J. Seo, "Robotic edge-rolling manipulation: A grasp planning approach," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3137–3144, Oct. 2018.

[9] S. Kolathaya, W. Guffey, R. W. Sinnet, and A. D. Ames, "Direct collocation for dynamic behaviors with nonprehensile contacts: Application to flipping burgers," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3677–3684, Oct. 2018.

[10] H. Yang, M. Xu, W. Li, and S. Zhang, "Design and implementation of a soft robotic arm driven by SMA coils," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6108–6116, Aug. 2019.

[11] T. Kishi, S. Shimomura, H. Futaki, H. Yanagino, M. Yahara, S. Cosentino, T. Nozawa, and K. Hashimoto, "Development of a humorous humanoid robot capable of quick-and-wide arm motion," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 1081–1088, Jul. 2016.
[12] M. Makarov, M. Grossard, P. Rodríguez-Ayerbe, and D. Dumur, "Modeling and preview H_∞ control design for motion control of elastic-joint robots with uncertainties," *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6429–6438, Oct. 2016.
[13] Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, and L. Ding, "Adaptive impedance control of human–robot cooperation using reinforcement learning," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8013–8022, Oct. 2017.
[14] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, "Multiobjective optimization for stiffness and position control in a soft robot arm module," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 108–115, Jan. 2018.
[15] A. Suarez, G. Heredia, and A. Ollero, "Physical-virtual impedance control in ultralightweight and compliant dual-arm aerial manipulators," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2553–2560, Jul. 2018.
[16] R. Weitschat and H. Aschemann, "Safe and efficient human–robot collaboration Part II: Optimal generalized human-in-the-loop real-time motion generation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3781–3788, Oct. 2018.
[17] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4249–4256, Oct. 2018.
[18] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Constraint-based and sensorless force control with an application to a lightweight dual-arm robot," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 340–347, Jan. 2016.
[19] S. C. Wang and Y. H. Liu, "A PSO-based fuzzy-controlled searching for the optimal charge pattern of Li-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2983–2993, May 2015.
[20] Z. Ma, Y. Dong, H. Liu, X. Shao, and C. Wang, "Forecast of non-equal interval track irregularity based on improved grey model and PSO-SVM," *IEEE Access*, vol. 6, pp. 34812–34818, 2018.
[21] C. Mao, R. Lin, C. Xu, and Q. He, "Towards a trust prediction framework for cloud services based on PSO-driven neural network," *IEEE Access*, vol. 5, pp. 2187–2199, 2017.
[22] L. Tong, X. Li, J. Hu, and L. Ren, "A PSO optimization scale-transformation stochastic-resonance algorithm with stability mutation operator," *IEEE Access*, vol. 6, pp. 1167–1176, 2017.
[23] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, "A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1496–1509, Jun. 2017.
[24] T.-H. S. Li, C.-Y. Liu, P.-H. Kuo, N.-C. Fang, C.-H. Li, C.-W. Chen, C.-Y. Hsieh, L.-F. Wu, J.-J. Liang, and C.-Y. Chen, "A three-dimensional adaptive PSO-based packing algorithm for an IOT-based automated e-fulfillment packaging system," *IEEE Access*, vol. 5, pp. 9188–9205, 2017.
[25] H. H. Chou, L. Y. Hsu, and H. T. Hu, "Turbulent-PSO-based fuzzy image filter with no-reference measures for high-density impulse noise," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 296–307, Feb. 2013.
[26] S.-K. Chou, M.-K. Jiau, and S.-C. Huang, "Stochastic set-based particle swarm optimization based on local exploration for solving the carpool service problem," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1771–1783, Aug. 2016.
[27] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73.
[28] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. Congr. Evol. Comput.*, vol. 1. 2001, pp. 94–100.
[29] M. Basu, "Artificial immune system for dynamic economic dispatch," *Int. J. Elect. Power Energy Syst.*, vol. 33, no. 1, pp. 131–136, 2011.
[30] Y.-L. Zheng, L.-H. Ma, L.-Y. Zhang, and J.-X. Qian, "Empirical study of particle swarm optimizer with an increasing inertia weight," in *Proc. Congr. Evol. Comput. (CEC)*, Dec. 2003, pp. 221–226.
[31] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3658–3670, Jun. 2011.
[32] T.-H. S. Li, C. J. Lin, P. H. Kuo, and Y. H. Wang, "Grasping posture control design for a home service robot using an ABC-based adaptive PSO algorithm," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 3, p. 118, 2016. doi: 10.5772/64044.
[33] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, Oct. 1995, pp. 39–43.
[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4. Nov./Dec. 1995, pp. 1942–1948.
[35] *Experimental Video*. Accessed: Feb. 3, 2019. [Online]. Available: https://youtu.be/tgD7OVJKcfc

TZUU-HSENG S. LI (S'85–M'90) received the B.S. degree from the Tatung Institute of Technology, Taipei, Taiwan, in 1981, and the M.S. and Ph.D. degrees from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1985 and 1989, respectively, all in electrical engineering. Since 1985, he has been with the Department of Electrical Engineering, NCKU, where he is currently a Distinguished Professor. From 1996 to 2009, he was also a Researcher with the Engineering and Technology Promotion Center, National Science Council, Tainan. From 1999 to 2002, he was the Director of the Electrical Laboratories with NCKU. From 2009 to 2012, he was the Dean of the College of Electrical Engineering and Computer Science, National United University, Miaoli City, Taiwan. From 2009 to 2016, he was the Vice President of the Federation of International Robot-Soccer Association. Since 2014, he has been the Director of the Center for Intelligent Robotics and Automation, NCKU. His current research interests include artificial and biological intelligence and applications, fuzzy system and control, home service robots, humanoid robots, mechatronics, 4WIS4WID vehicles, and singular perturbation methodology. Dr. Li was a recipient of the Outstanding Automatic Control Award from the Chinese Automatic Control Society (CACS), Taiwan, in 2006, and the Outstanding Research Award from the Ministry of Science and Technology, Taiwan, in 2017. He was a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor of the *Asia Journal of Control*. He is currently an Editor-in-Chief of *iRobotics*, and an Associate Editor of the *International Journal of Electrical Engineering*, the *International Journal of Fuzzy Systems*, and the IEEE TRANSACTIONS ON CYBERNETICS. He was elected as the President of the CACS, from 2008 to 2011, and the Robotics Society of Taiwan, from 2012 to 2015. He was elevated to CACS Fellow and RST Fellow, in 2008 and 2018, respectively.



PING-HUAN KUO received the B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2008, 2010, and 2015, respectively. Since 2017, he has been with the Computer and Intelligent Robot Program for Bachelor Degree, National Pingtung University (NPTU), where he is currently an Assistant Professor. His current research interests include fuzzy control, intelligent algorithms, humanoid robot, image processing, robotic application, big data analysis, machine learning, and deep learning applications.

YA-FANG HO was born in Kaohsiung, Taiwan, in 1988. She received the B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2010, 2011, and 2017, respectively. Her current research interests include fuzzy control, intelligent systems, humanoid robot, image processing, robotic applications, and FIRA/RoboCup games.

GUAN-HONG LIOU received the B.S. degree from the Department of Electronic Engineering, Tamkang University, New Taipei City, Taiwan, in 2013, and the M.S. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2016. His current research interests include fuzzy control, intelligent systems, humanoid robot, image processing, robotic applications, and FIRA/RoboCup games.