

Article

A Modified Biogeography-Based Optimization Approach for Visual Position-Based Inverse Kinematics of Robotic Arms

Liancheng Zheng^{1,2}, Mohammad Soleimani Amiri^{3,*} , Rizauddin Ramli^{2,*} , Sharifah Sakinah Syed Ahmad³ and Xiaotian Ma¹

¹ School of Mechanical Engineering, Shandong Huayu University of Technology, Dezhou 253034, China; p159501@siswa.ukm.edu.my (L.Z.); maxiaotian10@gmail.com (X.M.)

² Department of Mechanical and Manufacturing Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia

³ Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Durian Tunggal 76100, Melaka, Malaysia; sakinah@utem.edu.my

* Correspondence: soleimani@utem.edu.my (M.S.A.); rizauddin@ukm.edu.my (R.R.)

Abstract

Precise end-effector positioning of a robotic arm is critical for accurate performance in robotics, which directly impacts the system's performance in applications requiring high precision. This paper presents an optimized tag-based inverse kinematics approach for a 6 DoF robotic arm using a Modified Biogeography-Based Optimization (MBBO) algorithm, which is an enhanced version of the original Biogeography-Based Optimization (BBO), a population-based evolutionary algorithm inspired by the natural distribution of species across habitats. The target position is identified via AprilTag visual fiducial markers and integrated into the inverse kinematics solver. Performance was evaluated against Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and BBO through 3-dimensional simulations involving ten target points. Results show that MBBO achieves reduced positioning error compared to GA, PSO, and BBO, resulting in lower end-effector position errors. The findings highlight the effectiveness of combining visual tag detection with advanced optimization for precise robotic arm control.

Keywords: inverse kinematic; forward kinematic; biogeographical-based optimization; AprilTag

1. Introduction

Through the continual advancement of robotic technology, a variety of robot types are becoming increasingly integrated into our everyday lives, assisting humans in a wide range of applications [1,2]. Achieving precision in the end-effector positioning of a robotic arm is crucial in robotics. This can directly impact the system's performance in applications requiring high precision, such as assembly, surgery, and manipulation because it enhances the robot's efficiency, safety, and overall effectiveness in complex operational environments. The robotic arm is one of the most common types of robots that has been used in numerous industries like automotive, agriculture, and biomedicine. Its flexibility, strength, and accuracy make it ideal for handling complex tasks, such as avoiding joint limits and obstacles [3,4].

To replicate the intricate structure of the human arm, robotic arms are designed with manipulators made up of several segments and joints, generally divided into three main parts: the arm, wrist, and gripper [5]. These components allow robotic arms to



Academic Editors: Tuan Anh Luong and Babar Jamil

Received: 15 August 2026

Revised: 2 September 2026

Accepted: 3 September 2026

Published: 9 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

perform delicate operations with high precision. Moreover, integrating computer vision significantly enhances their capabilities by enabling them to perceive and interpret visual information [6]. Through computer vision, visual inputs such as images or videos are transformed into meaningful data, allowing robots to identify objects, monitor motion, and respond to changes in their surroundings [7]. In robotics, computer vision is extensively utilized across various domains, including automotive manufacturing, medical technology, and automated inspection [8]. By incorporating computer vision algorithms, robotic systems can process real-time data to accurately control their movements, a capability essential for tasks involving object recognition and manipulation [9].

The utilization of fiducial marker detection systems is a precise method for acquiring the position and orientation of specific tags in relation to a camera, as documented in several studies [10–12]. Due to its cost-effectiveness, accessibility, and accuracy, this technique has gained prominence in drone applications. For instance, Tang et al. [13] developed an algorithm for drone applications, incorporating a range of sensors, including multiple cameras and a 2-D laser scanner, with AprilTag arrays serving as the calibration target. Zhao et al. [14] investigated how the number and spatial distribution of fiducial markers affect the accuracy of robot-guided implant placement in edentulous mandibular models. Their study demonstrated that marker quantity is a key factor for achieving high-precision robotic implant surgery, whereas marker placement strategy is less critical within the tested configurations.

Discovering the inverse kinematics solution is a crucial factor in achieving precise robot control and end-effector positioning, ultimately enhancing the accuracy of robot path following [15–17]. Different solutions for inverse kinematics have been explored for the robotic arm. Xu et al. [18] presented a fusion of brain–computer interface and computer vision techniques to steer a robotic arm’s end-effector to a predefined position with the assistance of a depth camera. An experiment was conducted with a 6 Degree-of-Freedom (DoF) robotic arm that received an initial desired trajectory from the user’s brain signals, combined with point clouds produced by a laser-scanning sensor.

Biogeography-Based Optimization (BBO) is an evolutionary algorithm inspired by biogeography principles, used effectively for solving complex optimization problems including inverse kinematics and trajectory planning of robotic arms. It has advantages in global search capability and convergence speed compared to classical methods like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [19]. Serat et al. [20] explored artificial Intelligence-based computational methods, including the artificial immune system and multi-layer perceptron BBO, to calculate joint angles for a 6 DoF robotic arm. Their simulation results demonstrate that these approaches effectively handle the complexity of inverse kinematics, proving their reliability and practicality for robotic applications. Ren et al. [21] proposed a Hybrid BBO (HBBO) algorithm that combines BBO with differential evolution to efficiently solve the inverse kinematics problem of an 8 DoF redundant humanoid manipulator. By integrating a hybrid migration strategy and Gaussian mutation, the HBBO improves convergence speed, enhances population diversity, and achieves accuracy in minimizing end-effector position errors.

From a biomimetic perspective, the proposed optimization framework is motivated by the ability of natural ecosystems to achieve efficient distribution, adaptation, and survival of species under changing environmental conditions. BBO translates these ecological principles into a computational search mechanism by representing candidate solutions as habitats and modeling the migration of information among habitats through immigration and emigration processes. In the present study, this biological analogy is extended through the Modified BBO (MBBO), in which adaptive migration parameters, mutation, and elite preservation are employed to reproduce the exploration, adaptation, and diversity-

maintenance characteristics observed in natural ecosystems. These biomimetic principles are particularly suitable for the inverse kinematics of robotic manipulators, where multiple feasible joint configurations must be explored while satisfying physical joint constraints and minimizing end-effector positioning error. Therefore, the proposed approach establishes a direct connection between biological adaptation mechanisms and robotic motion optimization, demonstrating how nature-inspired computational principles can be translated into an engineering solution for precise robotic positioning.

The main contribution of this paper is the development of an optimized solution for inverse kinematics by MBBO, in which the parameters of its operators are changed by increasing the iteration to provide a wider search for the global optimum during the initial iterations and narrow the search space while the algorithm reaches its final iterations. In addition, an AprilTag-based fiducial marker detection approach was employed to achieve precise end-effector positioning in a 6 DoF robotic arm.

Therefore, this study provided an effective solution for accurately determining the joint angles corresponding to the desired end-effector position. The combination of visual tag detection and advanced optimization enhanced both the accuracy and adaptability of the robotic arm's motion control.

The novelty of this paper is the development of the MBBO algorithm for solving position-based inverse kinematics for a 6 DoF robotic arm with AprilTag to detect the end-effector position. The contributions of this paper are as follows:

- An optimal position-based inverse kinematics had been developed based on MBBO for a 6 DoF robotic arm to ensure that the end-effector reaches the target point.
- The integration of the MBBO algorithm and Tag-based visual fiducial marker systems to detect the desired target position for the end-effector precisely.
- Simulations comparing the optimal position-based inverse kinematics solved by MBBO with other conventional optimization methods such as GA, PSO, and BBO.

The rest of the paper is structured as follows: Section 2 elaborates on the optimal position-based inverse kinematics and visual fiducial marker systems. Section 3 covers the results and discussion. Section 4 concludes this paper.

2. Optimal Position-Based Inverse Kinematics Solution

This section presents an optimal position-based inverse kinematics solution for a 6 DoF robotic arm using MBBO. The proposed method is integrated with a visual fiducial system to track the end-effector position in real time.

2.1. Forward Kinematic

The forward kinematics of the 6 DoF robotic arm involved finding the end-effector's position. This robotic arm was composed of a base, four links, a wrist, and a gripper, all connected in series through joints. It consisted of three rotary joints (1, 4, and 6) and three hinge joints (2, 3, and 5). Figure 1 represents the joints and kinematic structure of the robotic arm.

In this study, Denavit–Hartenberg (DH) algorithm was employed to systematically model the kinematic structure of the robotic arm. The DH algorithm provided a standardized method to define coordinate frames for each joint and link of the robotic arm, enabling precise representation of the spatial relationships between successive links. Table 1 details the specific DH parameters used the robotic arm including the link lengths, link twists, link offsets, and joint angles for each of the six joints.

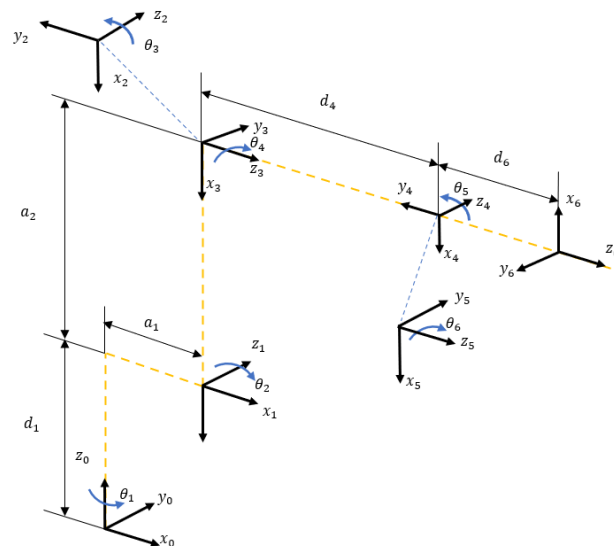


Figure 1. Kinematic structure of the robotic arm.

Table 1. DH parameters of the 6 DoF robotic arm.

Joints	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	$-\frac{\pi}{2}$
2	$\frac{\pi}{2} + \theta_2$	0	a_2	0
3	θ_3	0	0	$-\frac{\pi}{2}$
4	θ_4	d_4	0	$\frac{\pi}{2}$
5	θ_5	0	0	$-\frac{\pi}{2}$
6	θ_6	d_6	0	0

In Table 1, α_i , θ_i , d_i , and a_i denote the twist angle, joint angle, link offset, and link length, respectively. The transformation matrix of the end-effector regarding the base frame is determined by multiplying of the transformation matrix each frame to its previous frame, shown as follows:

$${}^0T_6 = {}^0T_1 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T \quad (1)$$

where ${}^0T_6 \in R^{4 \times 4}$ represents the transformation matrix of the end-effector with respect to the base frame. The x , y , and z positions of the end-effector in Cartesian coordinates relative to the reference frame are shown as follows:

$$x = C_1 a_1 + d_6 (C_5 C_1 S_{2'3} - S_5 (S_1 S_4 - C_4 C_1 C_{2'3})) + d_4 C_1 S_{2'3} - C_1 C_2 a_2 \quad (2)$$

$$y = S_1 a_1 + d_6 (C_5 S_1 S_{2'3} - S_5 (C_4 S_1 C_{2'3} + C_1 S_4)) + d_4 S_1 S_{2'3} - C_2 S_1 a_2 \quad (3)$$

$$z = d_1 + d_4 C_{2'3} + a_2 S_2 + d_6 (C_5 C_{2'3} - C_4 S_5 S_{2'3}) \quad (4)$$

where C_i and S_i denote the cosine and sine functions of the joint angle θ_i and C_i and S_{ij} represents cosine and sine of the joint angle $\theta_i + \theta_j$. Also, α_i , d_i , and a_i are determined based on the dimensions of the robotic arm, while θ_i is determined in real-time by an optimization algorithm.

In this paper, we used the ABB-IRB140 robotic arm, which is a compact and highly precise 6 DoF industrial manipulator widely utilized for assembly, handling, and research applications [22]. Based on the physical features of the ABB-IRB140 robotic arm, the DH

parameters in Table 1 are defined as follows: $a_1 = 70$ mm, $a_2 = 360$ mm, $d_1 = 352$ mm, $d_4 = 380$ mm and $d_6 = 65$ mm.

2.2. Optimal Position-Based Inverse Kinematic

This paper investigated position-based inverse kinematics as an optimization problem, with the goal of achieving the desired end-effector position by minimizing the objective function of the ABB-IRB140 robotic arm. Optimal position-based inverse kinematics refers to solving the position-based inverse kinematics problem through an optimization-based method rather than relying only on analytical equations. This approach is particularly important for 6 DoF robotic arm, where multiple joint configurations may satisfy the same end-effector position or where the robot may approach singularities and joint limits. By formulating the problem as an optimization task, the algorithm evaluates the solutions and selects the joint configuration that provides the best performance in terms of accuracy of the end-effector position. Therefore, the use of optimal position-based inverse kinematics ensures more reliable end-effector positioning and improves the overall behavior of the robotic arm during operation [23]. The objective function is formulated as the Euclidean position error between the current and desired end-effector positions, expressed as:

$$E_p = \sqrt{e_x^2 + e_y^2 + e_z^2} \quad (5)$$

where e_x , e_y , and e_z are the tracking error of the end-effector position represented as follows:

$$e_x = x - x_d \quad (6)$$

$$e_y = y - y_d \quad (7)$$

$$e_z = z - z_d \quad (8)$$

where x_d , y_d , and z_d represent the target end-effector positions in relation to the reference frame. Meanwhile, x , y , and z represent the end-effector current position, determined using forward kinematics.

The position-based inverse kinematic formulation was selected because the objective of the present study is point-to-point positioning of the robotic-arm end-effector, where the desired target is specified by its Cartesian position obtained from the AprilTag detection system. Therefore, the optimization objective focuses on minimizing the translational position error in the x , y , and z directions. Although a 6 DoF manipulator generally provides both position and orientation control, orientation constraints are not required for the point-to-point positioning task considered in this study. The resulting six joint angles provide a valid robot configuration for reaching the desired Cartesian target position while satisfying the joint-angle limits.

Various metaheuristic algorithms such as PSO and GA can be used for the optimization problem [24,25]. In this study, the MBBO was selected to solve the inverse kinematics, which is a nature-inspired optimization algorithm that draws inspiration from the study of biogeography based on the distribution of species and ecosystems across geographical space and time [26].

MBBO is an improved version of the conventional BBO, which is a population-based evolutionary algorithm inspired by the natural distribution of species across habitats. It enhances the original approach through mutation as an extended migration mechanism and by introducing dynamic operator adaptation, which helps to prevent premature convergence and avoid local optima.

Unlike the conventional BBO, the proposed MBBO incorporates several modifications to improve the search capability of the optimization process. First, a mutation mecha-

nism is introduced as an extended migration operation to increase population diversity. This mechanism promotes exploration of new regions of the search space and reduces the likelihood of the population becoming trapped in local optima. Second, adaptive parameters α_1 and α_2 are employed to dynamically adjust the optimization operators during the search process. This dynamic adaptation provides a balance between exploration and exploitation, allowing the algorithm to explore promising regions during the early stages while increasingly exploiting high-quality solutions as the optimization progresses. Third, an elite preservation strategy is incorporated to retain high-quality solutions between iterations. By protecting the best solutions from being lost during population updates, elite preservation strengthens exploitation and supports stable convergence toward an optimal or near-optimal solution. Therefore, the combination of mutation, adaptive operator adjustment, and elite preservation enables MBBO to improve population diversity, balance exploration and exploitation, avoid premature convergence, and achieve more stable convergence than conventional BBO.

The initial value of the MBBO was chosen randomly, and the number of populations or ‘habitats’ was usually determined by the designer. Although various methods, such as the Taguchi experimental design method, have been used to determine the population size in evolutionary algorithms, this paper took a trial-and-error approach and set the population size at 40. The population in the proposed MBBO is given as follows:

$$\mathbf{x}_{i,j} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} & \theta_{16} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} & \theta_{25} & \theta_{26} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} & \theta_{35} & \theta_{36} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_{40,1} & \theta_{40,2} & \theta_{40,3} & \theta_{40,4} & \theta_{40,5} & \theta_{40,6} \end{bmatrix} \quad (9)$$

where $i = 1, 2, \dots, 40$ is the number of habitats, and $j = 1, 2, \dots, 6$ is suitability index variable, which are design variables for MBBO, i.e., joint angles. Afterward, the habitats were evaluated by calculating the objective function and then sorted in ascending order. 20% of the habitats, referred to as the elites, were kept unchanged, while the remaining habitats changed migration and mutation processes to maintain diversity. This strategy ultimately improved the MBBO’s ability to discover global optima and prevented it from becoming trapped in local optima. The migration involved emigration and immigration rates, as illustrated in Figure 2.

In Figure 2, λ and μ represent immigration and emigration rates assigned to each individual in habitats. The initial habitat displays the lowest emigration rate and the highest immigration rate, creating a paradox where emigration is at its highest, while immigration is at its lowest. The migration operator is implemented by generating a random value and comparing it with the λ of all habitats. If the generated random number, r is smaller than λ , the migration operator is employed to establish new habitats for the upcoming iteration. Otherwise, the inhabitants of the previous generation remain unaltered. The migration operator to update the habitat for the next iteration is inspired by blend crossover, represented as follows:

$$\text{if } r_1 < \lambda_i: \quad x_{i,j}^{t+1} = \omega \left(\left(x_{k,j}^t + \alpha_1 \right) - \left(x_{i,j}^t \right) \alpha_2 \right), \quad j = 1, 2, \dots, 6 \quad (10)$$

where r_1 is a uniformly distributed random number in the range $(0, 1)$ and t is the iteration number. ω , α_1 , and α_2 are the parameters of the migration operator. $x_{k,j}$ is a selected habitat for creating new offspring habitat for the new iteration. The selection of k is determined using a roulette wheel scheme based on migration probabilities, as given by

$$p_i = \frac{\mu_i}{\sum_{n=1}^{40} \mu_n}, \quad i = 1, 2, \dots, 40, \quad (11)$$

where p_i denotes the migration probability of the (i)-th habitat. The cumulative migration probability is given as follows:

$$C_i = \sum_{n=1}^i p_n, \quad i = 1, 2, \dots, 40. \quad (12)$$

A random value is chosen from the roulette wheel and compared with each element in C_i . The first element of C_i that surpasses the random value is denoted as k , as defined in Equation (10). The roulette wheel algorithm increases the likelihood of selecting habitats with a higher emigration rate.

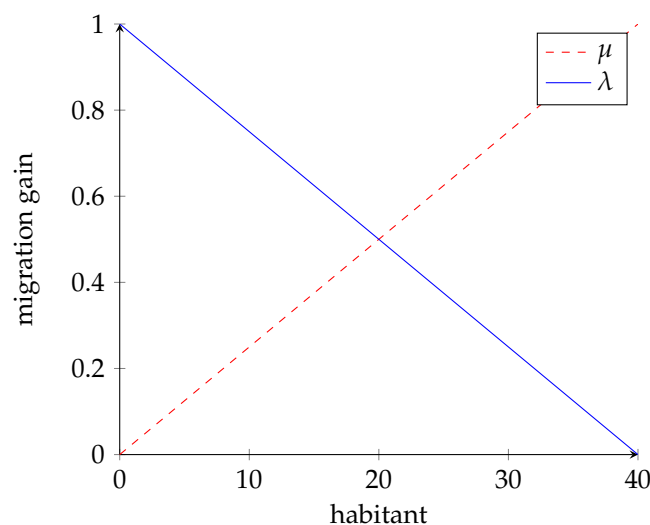


Figure 2. Migration rates in the population.

The parameters of the migration operator, including ω , α_1 , and α_2 , are not constant throughout all iterations. This adaptive behavior initially broadens the exploration of the search space during the early iterations and progressively narrows it down toward the global optimum as the number of iterations increases, as follows:

$$\omega = \omega_{min} + (w_{int} - w_{min})e^{-3a} \quad (13)$$

$$\alpha_1 = 2.3 \cdot b + 2.6 \quad (14)$$

$$\alpha_2 = 2.3 \cdot a + 0.1 \quad (15)$$

where ω_{int} is the initial parameters for ω , alternatively. b and a are dynamic parameters that vary within the range $[0, 1]$, with b decreasing and a increasing over iterations, as described below:

$$a = \frac{t}{t_{max}}, \quad \text{for } t = 1, 2, \dots, t_{max} \quad (16)$$

$$b = 1 - a \quad (17)$$

where t_{max} represents the maximum iteration value. The ω_{int} and ω_{min} are set at 1.2 and 0.3. This provides gradual refinement around the optimum for Equation (13).

This design enables the MBBO to explore a broad search space during the early iterations and gradually narrow its focus toward the global optimum in later iterations.

Such an adaptive strategy enhances the algorithm's performance by increasing early-stage iterations to prevent local optima convergence and achieve more precise solutions.

The sum of α_1 and α_2 remains constant at 5 throughout all iterations, with α_1 decreasing and α_2 increasing. Figure 3 illustrates the variations in migration parameters as the number of iterations changes. The maximum number of iterations is set to 500.

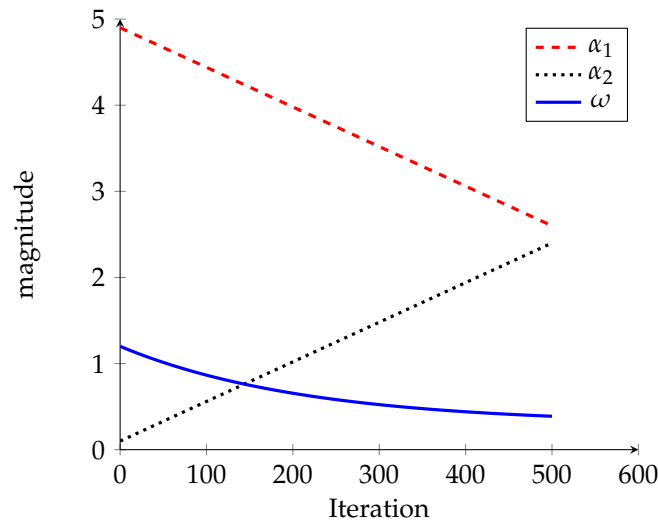


Figure 3. Changes in ω , α_1 , and α_2 across all iterations.

In addition to the migration process, the mutation operator plays a crucial role in generating new solutions within the MBBO framework. The mutation probability, m_p , is a predefined positive constant less than one, given as follows:

$$m_p \in [0, 1] \quad (18)$$

A randomly generated value, r , which is applied in each iteration, is compared with m_p , and when it is less than r , mutation is applied to the selected habitat. The mutation is mathematically defined as follows:

$$\text{if } r_2 < m_p : x_{k,j}^{t+1} = x_{i,j}^t + \sigma_j, \quad \text{for } j = 1, 2, \dots, 6 \quad (19)$$

where r_2 is a uniformly distributed random number in the range $(0, 1)$. σ_j is independently generated randomly for each joint variable according to a uniform distribution. Therefore, the mutation operator can either increase or decrease the corresponding joint angle, thereby maintaining population diversity and eliminating directional bias during the search process. $x_{k,j}$ is the selected habitat by the roulette wheel scheme to apply the mutation.

Since the optimization variables correspond to the robot joint angles, all candidate solutions are constrained within the physical joint limits of the ABB-IRB140 robotic arm. After each migration and mutation operation, a boundary-check mechanism is applied to ensure that every joint variable satisfies $(\theta_j^{\min} \leq \theta_j \leq \theta_j^{\max})$. Any variable exceeding its upper or lower bound is clipped to the corresponding limit. This guarantees that all habitats represent feasible robot configurations throughout the optimization process. The lower and upper bounds of each joint are determined according to the mechanical motion limits of the ABB-IRB 140 robotic manipulator, as listed in Table 2. During the optimization process, any candidate solution is limited to the corresponding boundary to ensure physically feasible joint configurations.

After the migration, mutation, and angle boundary-check operations, all offspring habitats are evaluated using the objective function and sorted in ascending order. Let

$$\mathbf{x}^{off} = \{\mathbf{x}_1^{off}, \mathbf{x}_2^{off}, \dots, \mathbf{x}_i^{off}\}, \quad (20)$$

denote the offspring population after sorting, where $\mathbf{x}_1^{off}$ represents the best habitat. Likewise, the elite habitats preserved from the previous generation are defined as

$$\mathbf{e}^t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_{i_e}^t\}, \quad (21)$$

where i_e denotes the number of elite habitats. The new population is then obtained by replacing the worst i_e offspring habitats with the elite habitats from the previous generation according to

$$\mathbf{x}_k^{t+1} = \begin{cases} \mathbf{x}_r^{off}, & r = 1, 2, \dots, i - i_e, \\ \mathbf{x}_{r-(i-i_e)}^t, & r = i - i_e + 1, \dots, i. \end{cases} \quad (22)$$

Algorithm 1 shows the pseudo code of MBBO.

Algorithm 1 Pseudo code of MBBO

```

1: Start;
2: Generate the initial population of 40 habitats;
3: Evaluate all habitats;
4: Sort the habitats in ascending order;
5: Set the iteration counter  $t = 1$ ;
6: Set initial parameters for  $r$ ,  $\mu$ ,  $\lambda$ , and  $m_p$ ;
7: while  $t < t_{\max}$  do
8:   Store the top 20% habitats as elite solutions;
9:   for  $i = 1$  to 40 do
10:    Generate a random number  $r \in (0, 1)$ ;
11:    if  $r_1 < \lambda_i$  then
12:      Select habitat  $k$  using roulette wheel selection;
13:      Compute  $\omega$ ,  $\alpha_1$ , and  $\alpha_2$ ;
14:      Update habitat by migration operator:
15:       $x_{i,j}^{t+1} = \omega((x_{k,j}^t + \alpha_1) - (x_{i,j}^t)\alpha_2)$ ;
16:    end if
17:    if  $r_2 < m_p$  then
18:      Mutate habitat according to:
19:       $x_{k,j}^{t+1} = x_{i,j}^t + \sigma_j$ ;
20:    end if
21:    Apply angle boundaries:
22:     $x_{i,j} = \max(\theta_j^{\min}, \min(x_i, \theta_j^{\max}))$ ;
23:  end for
24:  Evaluate all habitats;
25:  Sort the habitats in ascending order;
26:  Replace the worst habitats with the elite habitats;
27:  Update the current population with the modified habitats;
28:   $t = t + 1$ ;
29: end while
30: Select the top-ranked habitat as the best solution;
31: End;

```

The maximum iteration $t_{\max}$ is 500. In this algorithm, the term $(x_{k,j}^t + \alpha_1)$ is the scalar combination inside the habitat of the migration operator. The significance of MBBO compared to the conventional BBO is that its operation parameters vary by iteration, as mentioned in Figure 3 and Equations (13)–(15). This provides a wider search for the global

optima during the initial iterations and narrows the search space while the algorithm reaches its final iterations. The advantages of MBBO are the faster response and lower chances of being trapped in local optima. This also provides lower positioning error for the final result.

Table 2. Joint angle limits of the ABB-IRB 140 robotic manipulator.

Joint	Lower Bound (°)	Upper Bound (°)
θ_1	−180	180
θ_2	−90	110
θ_3	−230	50
θ_4	−200	200
θ_5	−115	115
θ_6	−400	400

2.3. Visual Fiducial Marker System

In this study, visual target detection was achieved by equipping the robotic arm with a camera rigidly attached to its end-effector. The camera continuously captured images of the workspace, where AprilTag fiducial markers are placed at target locations. The AprilTag detection algorithm processes these images in real-time, estimating the unique ID and precise 3-Dimensional (3D) position and orientation of each tag relative to the camera frame using established perspective-n-point techniques [13]. Knowing the fixed transformation between the camera and the end-effector, the detected tag pose was transformed from the camera frame to the robot's coordinate system, thereby defining the desired end-effector position for the manipulator. This spatial information served as the input for the inverse kinematics solver, which computes the corresponding set of joint angles needed to drive the end-effector to the detected tag pose. By integrating visual perception with kinematic modeling, the robotic arm can autonomously and accurately adjust its posture to reach visually identified targets, enabling robust and adaptable manipulation within simulated environments.

Numerous visual fiducial marker systems, including ArUco, ARTag, and ARToolKit, are widely used in various applications. Among them, AprilTag has gained popularity in robotics applications due to its robustness and flexibility. These systems are employed to detect and distinguish 2D markers from one another [27].

In this paper, the AprilTag marker of the 36H11 family is utilized. The raw camera image is processed to detect the position and orientation of the marker relative to the camera frame [28]. This process can be divided into two main components: the tag detector and the code system. The AprilTag system extracts unique ID information embedded in the marker and estimates the position of the marker relative to the camera [23].

Let $H_i \in \mathbb{R}^{3 \times 3}$ be the homography matrix corresponding to tag $i \in D$, where i is the marker ID from the set of detected tags D . The homography H_i projects a point $(x_{\text{tag}}, y_{\text{tag}}, 0)$ from the coordinate frame of tag i , denoted as $\{t_i\}$, to the pixel coordinate $(x_{\text{im}}, y_{\text{im}})$ in the image frame $\{\text{im}\}$ [29]. Figure 4 illustrates the coordinate representation of a point in both the AprilTag marker and the image frame.

The homography matrix H_i is represented as follows:

$$H_i = PE \quad (23)$$

where,

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & T_x \\ R_{21} & R_{22} & T_y \\ R_{31} & R_{32} & T_z \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

where $\mathbf{P} \in \mathbb{R}^{3 \times 4}$ is the perspective projection matrix, which is obtained through the camera calibration process. f_x and f_y are the focal lengths, while c_x and c_y are the principal points, typically located near the center of the image frame $\{im\}$. $\mathbf{E} \in \mathbb{R}^{4 \times 3}$ represents the extrinsic matrix, where R_{ij} denotes the elements related to rotation, and T_x , T_y , and T_z are the translation components. Generally, the extrinsic matrix $\mathbf{E}$ is a 4×4 matrix, but the third column is omitted for computational efficiency [30]. The relationship between the AprilTag marker frame and the image frame is given as follows:

$$\begin{bmatrix} x'_{im} \\ y'_{im} \\ z'_{im} \end{bmatrix} = H_i \begin{bmatrix} \frac{x_{tag}}{s_{tag,t}} \\ \frac{y_{tag}}{s_{tag,t}} \\ 1 \end{bmatrix} \quad (25)$$

The vector obtained in Equation (25) is represented in homogeneous image coordinates. The corresponding Cartesian image coordinates are obtained through homogeneous normalization by dividing the first two components by the third component, given as follows:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} \frac{x'_{im}}{z'_{im}} \\ \frac{y'_{im}}{z'_{im}} \end{bmatrix} \quad (26)$$

Here, $s_{tag,i}$ represents the half-size length of tag i . Since the AprilTag frame $\{t_i\}$ and the camera body frame $\{B\}$ are defined with different origins and axis orientations, the estimated tag pose must be transformed from the tag coordinate frame to the camera body frame through the corresponding rigid-body transformation [31].

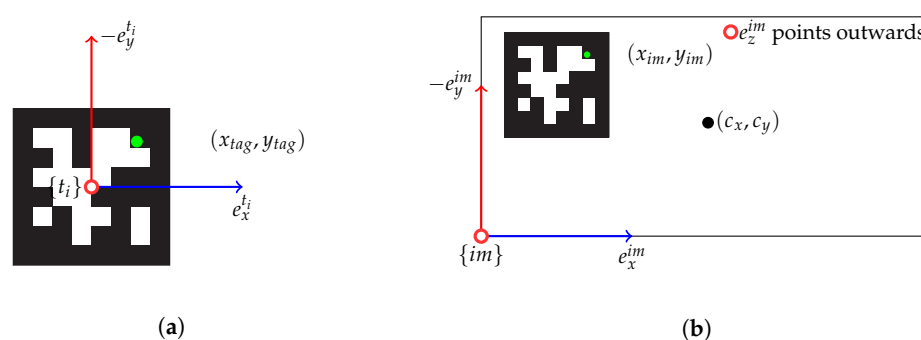


Figure 4. Coordination of a point in both tag marker and image frames: AprilTag Marker ID:1 (a) Image frame coordination (b) Image frame coordination.

3. Results and Discussion

To evaluate the performance of the optimal inverse kinematics using MBBO, a 3D model of the 6 DoF robotic arm is simulated within a virtual environment. The simulated robotic arm and camera view of the AprilTag are captured from a camera that is rigidly attached to the robotic arm end-effector is shown in Figure 5.

In this simulation, the robotic arm's end-effector was commanded to move from its target position to a target location identified by an AprilTag marker. The tag's position was accurately detected through the AprilTag vision system and established as the desired target

position for the task. This target position is then incorporated into the objective function of the optimal inverse kinematics problem, which is solved using the MBBO algorithm. The inverse kinematics solution yielded six joint angular positions, corresponding to the 6 DoF of the robotic manipulator. These angular positions served as control inputs applied to the simulated robotic arm within the virtual environment.

Since the robotic arm and the attached camera are simulated in a virtual environment, the camera parameters and camera-to-robot extrinsic transformation are defined directly within the simulation environment. Therefore, physical camera calibration and extrinsic calibration procedures are not performed in this study.

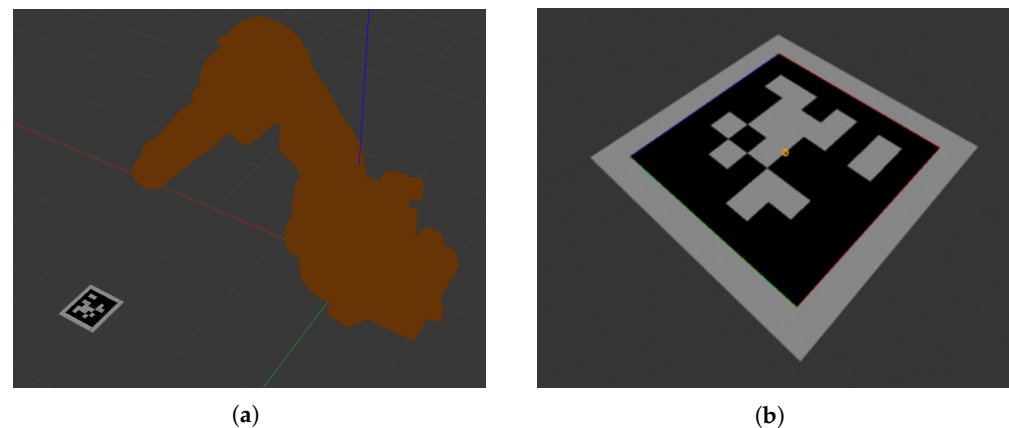


Figure 5. Simulated robotic arm in a virtual environment. (a) Robotic arm with AprilTag. (b) Camera View of AprilTag.

The camera is rigidly attached to the end-effector link; therefore, its position coincides with the end-effector position in the simulation. The AprilTag target position is initially obtained with respect to the camera coordinate frame and is transformed into the robotic-arm reference coordinate frame before being provided to the inverse kinematics solver. Since the camera is rigidly mounted on the end effector, the relative camera-to-end-effector transformation is fixed and predefined in the simulation.

Regarding AprilTag localization error, the present study evaluates the system in a virtual environment; therefore, real-world sources of localization error, such as image noise, illumination variations, lens distortion, camera resolution, and imperfect tag detection, were not experimentally evaluated. In the simulation, the detected AprilTag position is used as the target position for the inverse kinematics solver.

The proposed MBBO algorithm computes the optimal joint angles corresponding to the target position detected by the AprilTag system. These optimized joint angles are subsequently transmitted as reference commands to the ABB-IRB140 robotic model in the virtual environment. The robot then executes the corresponding motion while the joint positions are recorded over time. Figure 6 illustrates the time-domain joint responses of the ABB-IRB140 robotic arm during the execution of the optimized inverse kinematics solution for four different target positions detected by the AprilTag system in virtual environment.

From Figure 6 it is concluded that the optimal inverse kinematics by MBBO and integrated with AprilTag performed effectively for the robotic arm inverse kinematics solution. It is found that joints 1, 2, 3, and 5 exhibit significant motion, with joint 5 reaching approximately 1.3 radians and remaining constant thereafter, indicating a stable target position has been achieved. Joint 1 stabilizes near 0.8 radians, while joint 3 moves slightly into the negative range before settling, reflecting precise adjustments for accurate end-effector positioning. Joints 4 and 6, in contrast, remain near zero throughout the motion, suggesting minimal or no movement required for this particular task. Figure 6

demonstrated the effectiveness of the MBBO algorithm in generating feasible and stable joint angle solutions that enable the robotic arm to reach the target position identified by the AprilTag with precision and without abrupt transitions or oscillations.

In this study, 10 different target points for the end-effector are applied to the simulated robotic arm model and the optimal inverse kinematic, which is solved by GA, PSO, and MBBO for better comparison. To ensure a fair comparison among the optimization algorithms, identical population sizes and maximum iteration numbers were employed for all methods. The hyperparameter settings used in this study are summarized in Table 3.

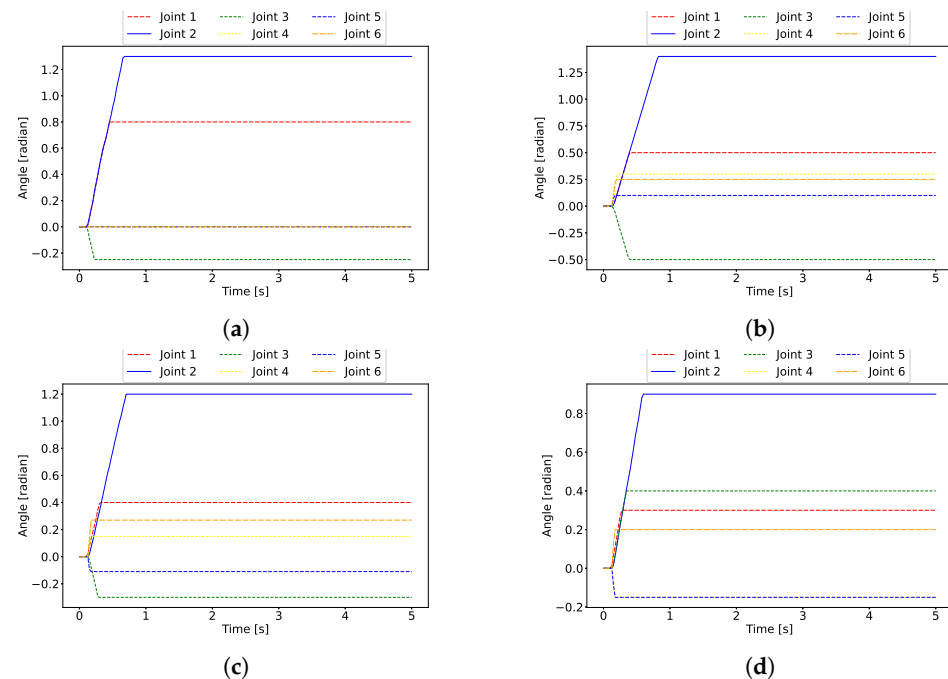


Figure 6. Time-domain joint responses generated by applying the optimal inverse kinematics solution obtained using the proposed MBBO algorithm for four different target points in virtual environment. (a) Point A, (b) Point B, (c) Point C, (d) point D.

Table 3. Hyperparameter settings for optimization algorithms.

Algorithm	Hyperparameters
GA	Population size = 40, Maximum iterations = 500, Crossover rate = 0.8, Mutation probability = 0.1
PSO	Swarm size = 40, Maximum iterations = 500, Inertia weight $w = 0.7$, Cognitive coefficient $c_1 = 2$, Social coefficient $c_2 = 2$
BBO	Population size = 40, Maximum iterations = 500, Mutation probability = 0.1
MBBO	Population size = 40, Maximum iterations = 500, Elite preservation = 20%, Mutation probability $m_p = 0.1$, Initial $\omega = 1.2$, Adaptive α_1 and α_2

Table 4 compares the objective function for MBBO, GA, PSO and BBO.

The optimal inverse kinematics solution was evaluated using GA, PSO, BBO, and MBBO across ten target points for the robotic arm end-effector. Since metaheuristic optimization algorithms exhibit stochastic behavior, each algorithm was independently executed 30 times for every target point. The results presented in Table 4 are reported as the mean and standard deviation of the Euclidean position error.

The MBBO algorithm achieved the lowest average position error among all compared optimization methods, demonstrating improved optimization accuracy and solution stability. Compared with GA, PSO, and conventional BBO, the MBBO reduced the average position error by approximately 67.3%, 76.1%, and 32%, respectively. Statistical analysis was further performed using the Wilcoxon signed-rank test to assess the significance of the differences between MBBO and the compared optimization algorithms across the ten target points. The results showed that MBBO achieved significantly the lowest position errors and $p = 0.00195$ for GA, PSO and BBO based-on Wilcoxon signed-rank test. These results indicate that the lower position errors achieved by MBBO were statistically significant across the evaluated target points. Therefore, the statistical analysis further supports the superior positioning accuracy of the proposed MBBO compared with the conventional optimization algorithms.

Table 4. Comparison of position error for GA, PSO, BBO, and MBBO over 30 independent runs.

Target Point (m)	GA	PSO	BBO	MBBO
A = [0.45, 0.10, 0.50]	0.032 ± 0.008	0.088 ± 0.021	0.021 ± 0.006	0.017 ± 0.004
B = [0.30, −0.20, 0.60]	0.036 ± 0.009	0.110 ± 0.028	0.024 ± 0.005	0.019 ± 0.005
C = [0.20, 0.35, 0.55]	0.058 ± 0.014	0.031 ± 0.010	0.023 ± 0.007	0.015 ± 0.004
D = [0.40, −0.30, 0.70]	0.042 ± 0.011	0.058 ± 0.017	0.028 ± 0.008	0.022 ± 0.006
E = [0.35, 0.40, 0.45]	0.086 ± 0.019	0.066 ± 0.016	0.031 ± 0.009	0.018 ± 0.005
F = [0.25, −0.45, 0.40]	0.062 ± 0.015	0.090 ± 0.023	0.020 ± 0.006	0.014 ± 0.004
G = [0.50, 0.15, 0.35]	0.035 ± 0.010	0.031 ± 0.009	0.019 ± 0.005	0.013 ± 0.003
H = [0.40, −0.35, 0.30]	0.045 ± 0.012	0.033 ± 0.011	0.018 ± 0.005	0.012 ± 0.003
I = [0.38, −0.42, 0.32]	0.044 ± 0.011	0.120 ± 0.030	0.022 ± 0.006	0.016 ± 0.004
J = [0.30, 0.45, 0.28]	0.048 ± 0.013	0.046 ± 0.012	0.021 ± 0.007	0.018 ± 0.005
K = [0.07, 0.00, 1.092]	0.071 ± 0.017	0.098 ± 0.025	0.034 ± 0.010	0.021 ± 0.006
L = [0.765, 0.00, 0.099]	0.064 ± 0.015	0.082 ± 0.021	0.029 ± 0.008	0.019 ± 0.005
M = [−0.290, 0.000, 0.609]	0.053 ± 0.026	0.075 ± 0.015	0.038 ± 0.016	0.017 ± 0.004
Average Error (m)	0.052 ± 0.021	0.071 ± 0.034	0.025 ± 0.010	0.017 ± 0.005

The integration of AprilTag detection to identify the target position of the robotic arm's end-effector significantly enhances the accuracy of the inverse kinematics, solved by MBBO. This improved target localization directly translates to more precise determination of joint angles, enabling the robotic arm to achieve highly accurate and reliable positioning of each joint. Consequently, the fusion of AprilTag-based visual feedback with inverse kinematics not only strengthened the overall control performance but also facilitated precise and adaptive manipulation tasks within the simulated environment. In addition, the performance of the MBBO compared to the other two conventional optimization problems showed that the MBBO provided the efficient solution for the inverse kinematics problem.

With the employment of AprilTag and the use of MBBO in determining the robotic arm's end-effector position demonstrates its superior accuracy and efficiency compared to GA, PSO and BBO methods. This resulted in more precise joint angle solutions and improved overall control performance, confirming MBBO as a more effective approach for inverse kinematics in robotic manipulation tasks.

To further evaluate the optimization algorithms under challenging target conditions in Table 4, points K and L, which are near the workspace boundaries of the ABB IRB140, were introduced. Point K represents a near-vertical workspace-boundary condition, while Point L represents a highly extended configuration near the radial workspace boundary. Point M was considered as a challenging singular configuration corresponding to a wrist-singular condition with $\theta_5 = 0^\circ$.

Figure 7 illustrates a comparison of the convergence behavior of GA, PSO, BBO, and MBBO algorithms.

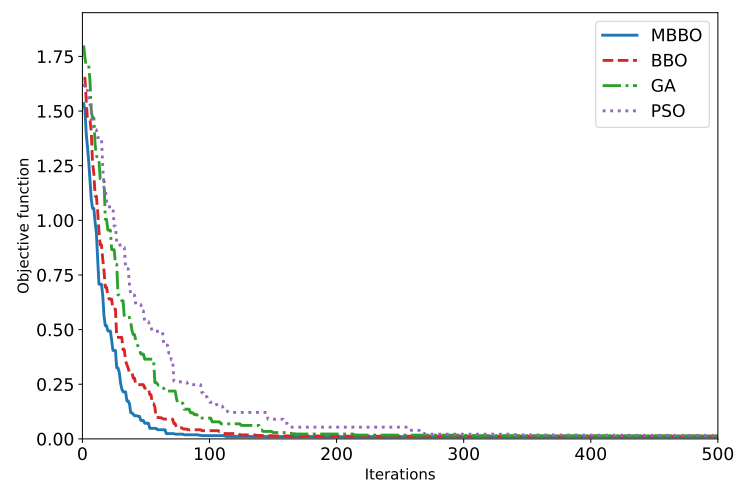


Figure 7. Comparison of convergence behavior of GA, PSO, BBO and MBBO algorithms.

MBBO demonstrates a faster reduction in the objective function value and reaches a lower final objective value compared with the other algorithms. This indicates that MBBO converges more rapidly toward high-quality solutions and provides more consistent optimization performance.

The proposed MBBO algorithm successfully solves the inverse kinematics problem using the target position obtained from the AprilTag detection system. Compared with GA, PSO, and the conventional BBO algorithm, MBBO achieves lower position errors and more accurate end-effector positioning across the tested target points.

4. Conclusions

In this paper, an optimal inverse kinematics solution for a 6 DoF robotic arm was presented. The position of the end-effector in the Cartesian system was developed by forward kinematics. The MBBO solved the inverse kinematics problem, and its operators determined dynamic adaptation across iterations to avoid trapping in local optima and achieve a more accurate solution.

The performance of the inverse kinematics by MBBO integrated with AprilTag was examined in a virtual environment to execute point-to-point motion to a specific target point for the robot end-effector. The presented method provided an accurate and reliable inverse kinematics solution that was integrated with a visual fiducial marker system. The comparison of the MBBO as the suggested optimization algorithm for inverse kinematics solution with GA, PSO, and BBO demonstrated superior positioning accuracy for solving this problem.

The limitation of this work is by not executing the MBBO to the actual robotic arm. In future work, the path planning algorithm can be added to this algorithm to avoid the end-effector collision with the possible obstacles.

Author Contributions: Conceptualization, M.S.A. and R.R.; methodology, M.S.A. and R.R.; software, M.S.A. and S.S.S.A.; validation, M.S.A., R.R. and L.Z.; formal analysis, M.S.A., R.R., L.Z. and X.M.; investigation, M.S.A., L.Z. and R.R.; resources, L.Z. and X.M.; data curation, M.S.A., S.S.S.A. and R.R.; writing—original draft preparation, M.S.A.; writing—review and editing, S.S.S.A., R.R. and M.S.A.; visualization, M.S.A.; supervision, R.R.; project administration, R.R.; funding acquisition, L.Z. and X.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Specialized Intelligent Manufacturing Engineering Laboratory for the financial support received under grant scheme PT2025KJS002 and Shandong Province Higher Education Institutions Marine Vessel Special Motor Key Technology Development and Component Manufacturing University-Enterprise Collaborative Innovation Center under grant scheme PT2025KJS005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM), Universiti Kebangsaan Malaysia (UKM) and Shandong Huayu University of Technology for supporting this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kanehiro, F.; Suleiman, W.; Griffin, R. Editorial: Humanoid Robots for Real-World Applications. *Front. Robot. AI* **2022**, *9*, 938775. [[CrossRef](#)] [[PubMed](#)]
2. Gholipour, R.; Fateh, M.M. Designing a robust control scheme for robotic systems with an adaptive observer. *Int. J. Eng. Trans. B Appl.* **2019**, *32*, 270–276. [[CrossRef](#)]
3. Tang, X.; Zhou, H.; Xu, T. Obstacle avoidance path planning of 6-DOF robotic arm based on improved A* algorithm and artificial potential field method. *Robotica* **2024**, *42*, 457–481. [[CrossRef](#)]
4. Jin, T.; Han, X. Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects. *Comput. Electron. Agric.* **2024**, *221*, 108938. [[CrossRef](#)]
5. Sukri, H.; Ibadillah, A.F.; Thinakaran, R.; Umam, F.; Dafid, A.; Kurniawan, A.; Morshed, M.M.; Kurniawan, D. Enhanced Precision Control of a 4-DOF Robotic Arm Using Numerical Code Recognition for Automated Object Handling. *J. Robot. Control* **2025**, *6*, 315–335. [[CrossRef](#)]
6. Abdi, A.; Ranjbar, M.H.; Park, J.H. Computer Vision-Based Path Planning for Robot Arms in Three-Dimensional Workspaces Using Q-Learning and Neural Networks. *Sensors* **2022**, *22*, 1697. [[CrossRef](#)] [[PubMed](#)]
7. Schuchert, P.; Karimi, A. High-precision control of a robotic arm using frequency-based data-driven methods. *Control Eng. Pract.* **2025**, *155*, 106175. [[CrossRef](#)]
8. Cohen, Y.; Biton, A.; Shoval, S. Fusion of Computer Vision and AI in Collaborative Robotics: A Review and Future Prospects. *Appl. Sci.* **2025**, *15*, 7905–7935. [[CrossRef](#)]
9. Phuc, T.D.; Son, B.C. Development of an autonomous chess robot system using computer vision and deep learning. *Results Eng.* **2025**, *25*, 104091. [[CrossRef](#)]
10. Adapa, D.; Shekhawat, V.S.; Gautam, A.; Mohan, S. Visual Assist System With Enhanced Localization for Indoor Navigation of Visually Impaired People. *IEEE Access* **2024**, *12*, 162075–162093. [[CrossRef](#)]
11. Vannozzi, L.; Guachi-Guachi, L.; Ruspi, J.; Ciancia, S.; Baldi, G.; Lunni, D.; Scarlino, P.; Poliziani, A.; Zucca, A.; Bellini, M.; et al. Automated Strategy for Tissue Analysis in Anatomic Pathology: Fiducial Marker Integration and Multisurface Tissue Comparison. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 11601–11612. [[CrossRef](#)]
12. Malyuta, D.; Brommer, C.; Hentzen, D.; Stastny, T.; Siegwart, R.; Brockers, R. Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *J. Field Robot.* **2019**, *37*, 137–157. [[CrossRef](#)]
13. Tang, D.; Hu, T.; Shen, L.; Ma, Z.; Pan, C. AprilTag array-aided extrinsic calibration of camera–laser multi-sensor system. *Robot. Biomim.* **2016**, *3*, 13. [[CrossRef](#)] [[PubMed](#)]
14. Zhao, Y.; Liao, Y.; Wu, X.; Zhang, Y.; Shi, B.; Yan, Q. Effect of the number and distribution of fiducial markers on robot-guided implant placement accuracy in edentulous mandibular phantoms. *J. Dent.* **2023**, *134*, 104529. [[CrossRef](#)] [[PubMed](#)]

15. Shah, S.K.; Mishra, R.; Samprit Ray, L. Solution and validation of inverse kinematics using Deep Artificial neural network. *Mater. Today Proc.* **2020**, *26*, 1250–1254. [[CrossRef](#)]
16. Shirafuji, S.; Ota, J. Kinematic Synthesis of a Serial Robotic Manipulator by Using Generalized Differential Inverse Kinematics. *IEEE Trans. Robot.* **2019**, *35*, 1047–1054. [[CrossRef](#)]
17. Kouabon, G.J.; Melingui, A.; Mvogo Ahanda, J.J.B.; Lakhal, O.; Coelen, V.; KOM, M.; Merzouki, R. A Learning Framework to inverse kinematics of high DOF redundant manipulators. *Mech. Mach. Theory* **2020**, *153*, 103978. [[CrossRef](#)]
18. Xu, Y.; Ding, C.; Shu, X.; Gui, K.; Bezsudnova, Y.; Sheng, X.; Zhang, D. Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance. *Robot. Auton. Syst.* **2019**, *115*, 121–129. [[CrossRef](#)]
19. Masdari, M.; Nabavi, S.S.; Ahmadi, V. An overview of virtual machine placement schemes in cloud computing. *J. Netw. Comput. Appl.* **2016**, *66*, 106–127. [[CrossRef](#)]
20. Serat, A.; Djebbar, B. Integrating artificial immune systems and multi-layer perceptron-biogeography-based optimization for enhanced inverse kinematics in robotic arm. *IAES Int. J. Robot. Autom. (IJRA)* **2024**, *13*, 401–409. [[CrossRef](#)]
21. Ren, Z.; Wang, Z.; Sun, L. A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-DOF redundant humanoid manipulator. *Front. Inf. Technol. Electron. Eng.* **2015**, *16*, 607–616. [[CrossRef](#)]
22. Fazilat, M.; Zioui, N. Quantum neural network-based inverse kinematics of a six-jointed industrial robotic arm. *Robot. Auton. Syst.* **2025**, *194*, 105123. [[CrossRef](#)]
23. Amiri, M.S.; Ramli, R. Utilisation of Initialised Observation Scheme for Multi-Joint Robotic Arm in Lyapunov-Based Adaptive Control Strategy. *Mathematics* **2022**, *10*, 3126. [[CrossRef](#)]
24. Qu, Z.; Younis, W.; Liu, X.; Junejo, A.K.; Almutairi, S.Z.; Wang, P. Optimized PID Controller for Load Frequency Control in Multi-Source and Dual-Area Power Systems Using PSO and GA Algorithms. *IEEE Access* **2024**, *12*, 186658–186678. [[CrossRef](#)]
25. Zhang, Z.; Wu, X.; Wang, B.; Liu, H.; Meng, J. Adaptive grid-forming strategy for a photovoltaic storage system based on edge transfer PSO algorithm. *Energy Rep.* **2024**, *12*, 1683–1692. [[CrossRef](#)]
26. Amiri, M.S.; Ramli, R. Offline tuning mechanism of joint angular controller for lower-limb exoskeleton with adaptive biogeographical-based optimization. *Turk. J. Electr. Eng. Comput. Sci.* **2022**, *30*, 1654–1676. [[CrossRef](#)]
27. Costa, G.M.; Petry, M.R.; Martins, J.G.; Moreira, A.P.G.M. Assessment of Multiple Fiducial Marker Trackers on Hololens 2. *IEEE Access* **2024**, *12*, 14211–14226. [[CrossRef](#)]
28. Abbas, S.M.; Aslam, S.; Berns, K.; Muhammad, A. Analysis and Improvements in AprilTag Based State Estimation. *Sensors* **2019**, *19*, 5480–5512. [[CrossRef](#)] [[PubMed](#)]
29. Amiri, M.S.; Ramli, R.; Faizal, A.H. Simultaneous Localization and Mapping and Tag-Based Navigation for Unmanned Aerial Vehicles. *Int. J. Integr. Eng.* **2023**, *15*, 225–232. [[CrossRef](#)]
30. Song, C.; Ho, S.W.; Huang, W.; Tansu, N. Camera-Based Positioning System for Battery-Less Retroreflective Tags. *IEEE Sen. J.* **2024**, *24*, 22867–22876. [[CrossRef](#)]
31. Amiri, M.S.; Ramli, R.; Zaidi, I.E.; Van, M. PID Controller for Unmanned Aerial Vehicle in Closed Environment Using Fiducial Marker Systems. *J. Adv. Manuf. Technol.* **2024**, *18*, 105–123.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.