BEE ALGORITHM BASED CONTROL DESIGN FOR TWO-LINKS ROBOT ARM SYSTEMS

RUSSUL A. KADHIM, MINA Q. KADHIM, HUTHAIFA AL-KHAZRAJI*, AMJAD J. HUMAIDI

Control and System Engineering Department, University of Technology-Iraq, Baghdad-Iraq

**Corresponding author: 60141@uotechnology.edu.iq*

(Received: 29 January 2024; Accepted: 24 April 2024; Published online: 15 July 2024)

ABSTRACT: This paper presents a comparative study between two advanced versions of the classical Proportional-Integral-Derivative (PID) controller including the Proportional Integral minus Proportional Derivative (PI-PD) controller and the Nonlinear Proportional Derivative (NPD) to manipulate the angular position of the two-link robot arm system and eliminate the effects of the load disturbances. The dynamic equations of the two-link robot arm system were obtained based on the Lagrange approach. To determine the best value of the adjustable coefficients of each controller, the tuning process was converted to an optimization problem. Then, the Bee Algorithm (BA) optimization technique was employed to find the best value of the adjustable coefficients of each controller. The computer simulation results based on MATLAB show the NPD-BA controller outperformed the PI-PD-BA controller in normal conditions. Furthermore, the NPD-BA demonstrated a substantial enhancement when a load disturbance was applied.

ABSTRAK: Kajian ini membentangkan perbandingan antara dua versi terkini pengawal klasik Keseimbangan-Pengkamiran-Terbitan (PID) termasuk pengawal Keseimbangan Pengkamiran tolak Keseimbangan Terbitan (PI-PD) dan Keseimbangan Terbitan Tak Linear (NPD) bagi memanipulasi kedudukan sudut sistem lengan robot dua pautan dan menghapuskan kesan gangguan beban. Persamaan dinamik sistem lengan robot dua pautan ini diperoleh berdasarkan pendekatan Lagrange. Bagi menentukan nilai terbaik pekali boleh laras setiap pengawal, proses penalaan ditukar kepada masalah pengoptimuman. Kemudian, teknik pengoptimuman Algoritma Lebah (BA) digunakan bagi mencari nilai terbaik bagi pekali boleh laras setiap pengawal. Dapatan simulasi komputer berdasarkan MATLAB menunjukkan pengawal NPD-BA mengatasi prestasi pengawal PI-PD-BA dalam keadaan normal. Tambahan, NPD-BA menunjukkan peningkatan yang ketara apabila gangguan beban digunakan.

KEYWORDS: *Control Design, Two-link Robot Arm System, Proportional Integral minus Proportional Derivative Controller, Nonlinear Proportional Derivative Controller, Bee Algorithm*

1. INTRODUCTION

The two-link robot arm system is a major component in the modern manufacturing industry [1]. Additionally, robotic arms can be used as medical solutions for persons that face difficulty in performing physical activities [2]. It is a nonlinear, high coupled, multi-input multi-output (MIMO) and time varying dynamics system. One of the major challenges in controlling the two-link robot arm system is the uncertainties due to the unknown loads that must be handled by the robot arm (i.e. pick and place tasks) [3]. In the context of control design, the two-link robot arm system can be modeled as a double pendulum system with two degrees of freedom where the equation of motion may be established by Lagrange equation

[4]. Due to its complex dynamic structure, the two-link robot arm system can be considered as a benchmark system for testing and evaluation of different control approaches [5]. For example, Guechi et al. [6] presented a comparative study between Model Predictive Control (MPC) and the Linear Quadratic (LQ) control based on the feedback linearization of the two-link robot arm system. It was observed that the performance of the MPC control outperforms the performance of the LQ control approach. In the same way, Mohammed and Eltayeb [5] compared the performance of the Proportional-Integral-Derivative (PID) controller and Sliding Mode Control (SMC). The outcomes of this study revealed that the performance of the SMC has a faster and more robust response compared to the performance of the PID controller. However, better control signal was observed by SMC. As an alternative control strategy, the Fuzzy Logic Controller (FLC) was implemented by [2]. Baccouch and Dodds [1] proposed a robust PID controller. Recently, Bendimrad [7] introduced a SMC approach for controlling the two-link robot arm system. Taking advantage of the simple structure of the PID and the robustness of the SMC, Long et al. [8] proposed a variable structure PID control method for the two-link robot arm system. The outcomes show that the proposed approach enhanced the speed of the convergence by more than 80% compared with the classical PID control method, while maintaining the same steady-state accuracy. In terms of intelligence controller, Shen [9] proposed a Fuzzy Neural Network (FNN) controller. The design parameters of the proposed control are optimized by combining the Particle Swarm Optimization (PSO) and backpropagation (BP) algorithm. The findings of the study indicate that the system has good tracking performance, good adaptability, and stability by applying the control scheme.

Unlike previous studies, this paper presents a comparative study between two versions of the classical PID controller named Proportional-Integral minus Proportional-Derivative (PI-PD) controller and Nonlinear Proportional-Derivative (NPD) controller for controlling the two-link robot arm system. These two control approaches can be considered as an improved version of the classical PID controller. As opposed to the trial and error method to find the right value of the adjusted parameters of each controller, various swarm optimization algorithms have been proposed in the literature to achieve an optimal performance of the controllers. Swarm optimization algorithms have provided a substantial improvement in the capabilities of solving multivariate, high dimensional engineering problems, and at the same time, it is easy to implement [10-12]. This paper introduces the Bee Algorithm (BA) to tune the two controllers based on the error performance index.

The rest of the paper is organized as follows: the mathematical model of the two-link robot arm system is presented in Section 2. In Section 3 and Section 4, the proposed controllers are introduced and the bee algorithm is given, respectively. The simulation results and discussions are given in Section 5. Section 6 contains the conclusion.

2. MATHEMATICAL MODEL

This section describes in detailed the mathematical model of the two-link robot arm system. For simplicity, the system can be represented as a double pendulum with two masses M_1 and M_2 connected by two weightless rigid rods of lengths L_1 and L_2 as shown in Fig. 1 [1].

The two-link robot arm system has two degrees of freedom represented by the angle that rotates around the origin (θ_1) and the angle that rotates at the endpoint of the first pendulum (θ_2). The two angles (θ_1 and θ_2) are outputs of the system and they are manipulated by the two input torques (τ_1 and τ_2) [1].

https://doi.org/10.31436/iiumej.v25i2.3188

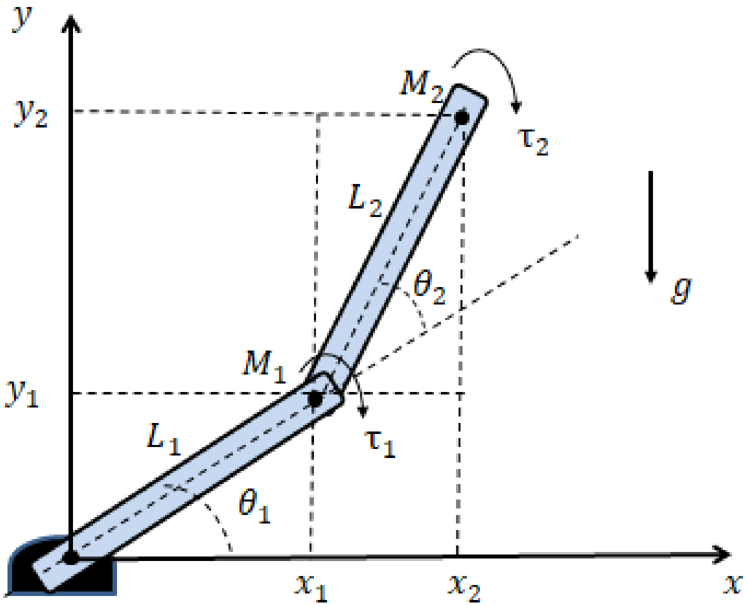

Figure 1. Two-link robot arm system

According to the Lagrangian approach to drive the equations of motion of the system, the Kinetic Energy (KE) and the Potential Energy (PE) of the system have to be obtained.

For the first mass, the equation of the mass in x direction and y direction is given by:

$$x_1 = L_1 cos\ (\theta_1) \tag{1}$$

$$y_1 = L_1 sin\ (\theta_1) \tag{2}$$

For the second mass, the equation of the mass in x direction and y direction is given by:

$$x_2 = L_1\ cos(\theta_1) + L_2 cos\ (\theta_2) \tag{3}$$

$$y_2 = L_1\ sin(\theta_1) + L_2 sin\ (\theta_2) \tag{4}$$

The velocity of the two masses is given by:

$$v_1 = \sqrt{\dot{x}_1{}^2 + \dot{y}_1{}^2} \tag{5}$$

$$v_2 = \sqrt{\dot{x}_2{}^2 + \dot{y}_2{}^2} \tag{6}$$

where

$$\dot{x}_1 = -L_1\dot{\theta}_1 sin\ (\theta_1) \tag{7}$$

$$\dot{y}_1 = L_1\dot{\theta}_1 cos\ (\theta_1) \tag{8}$$

$$\dot{x}_2 = -L_1\dot{\theta}_1\ sin(\theta_1) - L_2\dot{\theta}_2 sin\ (\theta_2) \tag{9}$$

$$\dot{y}_2 = L_1\dot{\theta}_1\ cos(\theta_1) + L_2\dot{\theta}_2 cos\ (\theta_2) \tag{10}$$

The Lagrangian equation is given by:

$$L = KE - PE \qquad (11$$

The kinematic energy of the system can be obtained as follows:

$$KE = \frac{1}{2}\ M_1\ \dot{v}_1 + \frac{1}{2}\ M_2\dot{v}_2 \tag{12}$$

https://doi.org/10.31436/iiumej.v25i2.3188

Substitute $\dot{v}_1$ and $\dot{v}_2$ as given in Eq. (5) and Eq. (6) respectively yields:

$$KE = \frac{1}{2} M_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} M_2 (\dot{x}_2^2 + \dot{y}_2^2 \tag{13}$$

Substitute $\dot{x}_1, \dot{y}_1, \dot{x}_2$ and $\dot{y}_2$ as given in Eq. (7), Eq. (8), Eq. (9) and Eq. (10), the KE can be rewritten as:

$$KE = \frac{1}{2} M_1 \left(\left(-L_1\dot{\theta}_1 sin\,(\theta_1)\right)^2 + \left(L_1\dot{\theta}_1 cos\,(\theta_1)\right)^2\right) + \frac{1}{2} M_2 \left(\left(-L_1\dot{\theta}_1\, sin(\theta_1) - L_2\dot{\theta}_2\, sin(\theta_2)\right)^2 + \left(L_1\dot{\theta}_1\, cos(\theta_1) + L_2\dot{\theta}_2\, cos(\theta_2)\right)^2\right) \tag{14}$$

Eq. (14) can be rearranged as follows:

$$KE = \frac{1}{2} (M_1 + M_2){L_1}^2{\dot{\theta}_1}^2 + \frac{1}{2} M_2{L_2}^2{\dot{\theta}_2}^2 + M_2L_1L_2\dot{\theta}_1\dot{\theta}_2\, cos(\theta_1 - \theta_2) \tag{15}$$

The potential energy of the system can be obtained as follows:

$$PE = M_1\, g\, y_1 + M_2\, g\, y_2 \tag{16}$$

Substitute y_1 and y_1 as given in Eq. (2) and Eq. (4) respectively obtains:

$$PE = M_1\, g\, (L_1 sin\,(\theta_1)\,) + M_2\, g\, (L_1\, sin(\theta_1) + L_2 sin\,(\theta_2)) \tag{17}$$

Eq. (17) can be simplified as:

$$PE = (M_1 + M_2)\, gL_1\, sin(\theta_1) + M_2 gL_2\, sin(\theta_2) \tag{18}$$

Substitute Eq. (15) and Eq. (18) into Eq. (11)

$$L = \left(\frac{1}{2} M_1 \left(\left(-L_1\dot{\theta}_1 sin\,(\theta_1)\right)^2 + \left(L_1\dot{\theta}_1 cos\,(\theta_1)\right)^2\right) + \frac{1}{2} M_2 \left(\left(-L_1\dot{\theta}_1\, sin(\theta_1) - L_2\dot{\theta}_2\, sin(\theta_2)\right)^2 + \left(L_1\dot{\theta}_1\, cos(\theta_1) + L_2\dot{\theta}_2\, cos(\theta_2)\right)^2\right)\right) - ((M_1 + M_2)\, gL_1\, sin(\theta_1) + M_2 gL_2\, sin(\theta_2)) \tag{19}$$

The Euler-Lagrange equation is determined as:

$$\frac{d}{dt}\left[\partial L/\partial\dot{\theta}_i\right] - \partial L/\partial\theta_i = \tau_i \qquad i = 1,2 \tag{20}$$

The partial derivatives of Eq. (20) $w.r.t\;\; to\; i = 1$ obtains:

$$\frac{\partial L}{\partial\dot{\theta}_1} = (M_1 + M_2)\, L_1^2\dot{\theta}_1 + M_2\, L_1L_2\, \dot{\theta}_2\, cos(\theta_1 - \theta_2 \tag{21}$$

$$\frac{\partial L}{\partial\theta_1} = -M_2\, L_1L_2\dot{\theta}_1\, \dot{\theta}_2\, sin(\theta_1 - \theta_2) - (M_1 + M_2)\, gL_1\, cos(\theta_1) \tag{22}$$

Then:

$$\frac{d}{dt}\left[\partial L/\partial\dot{\theta}_1\right] = (M_1 + M_2)\, L_1^2\ddot{\theta}_1 + M_2\, L_1L_2\, \ddot{\theta}_2\, cos(\theta_1 - \theta_2) - M_2\, L_1L_2\dot{\theta}_2\left(\dot{\theta}_1 - \dot{\theta}_2\right) sin(\theta_1 - \theta_2) \tag{23}$$

Substitute Eq. (22) and Eq. (23) into Eq. (20) $w.r.t\; i = 1$ obtains:

$$\left((M_1 + M_2)\, L_1^2\ddot{\theta}_1 + M_2\, L_1L_2\, \ddot{\theta}_2\, cos(\theta_1 - \theta_2) - M_2\, L_1L_2\dot{\theta}_2\left(\dot{\theta}_1 - \dot{\theta}_2\right) sin(\theta_1 - \theta_2)\right) - \left((M_1 + M_2)\, L_1^2\dot{\theta}_1 + M_2\, L_1L_2\, \dot{\theta}_2\, cos(\theta_1 - \theta_2)\right) = \tau_1 \tag{24}$$

In the same way, the partial derivatives of Eq. (20) $w.r.t\; to\;\; i = 2$ obtains:

$$\frac{\partial L}{\partial\dot{\theta}_2} = M_2L_2^2\dot{\theta}_2 + M_2\, L_1L_2\, \dot{\theta}_1\, cos(\theta_1 - \theta_2) \tag{25}$$

$$\frac{\partial L}{\partial \theta_2} = M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - M_2 g L_2 \cos(\theta_2) \tag{26}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_2}\right] = M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1 - \theta_2) \tag{27}$$

Substitute Eq. (26) and Eq. (26) into Eq. (20) $w.r.t\ i = 2$ obtains:

$$\left(M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1 - \theta_2)\right) - \left(M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - M_2 g L_2 \cos(\theta_2)\right) = \tau_2 \tag{28}$$

Solving Eq. (24) and Eq. (28) for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ respectively yields:

$$\ddot{\theta}_1 = g_1(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \tau_1, \tau_2) \tag{29}$$

$$\ddot{\theta}_2 = g_2(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \tau_1, \tau_2) \tag{30}$$

where

$$g_1 = \frac{\frac{M\ \tau_1}{M_2\ L_1} - M\ L_2 \dot{\theta}_2{}^2 \sin(\theta_1 - \theta_2) - g\cos(\theta_1) - M\ \cos(\theta_1 - \theta_2)\left[\frac{\tau_2}{M_2\ L_2} + L_1 \dot{\theta}_1{}^2 \sin(\theta_1 - \theta_2) - g\cos(\theta_2)\right]}{L_1(1 - M\cos^2(\theta_1 - \theta_2))} \tag{31}$$

$$g_2 = \frac{\frac{\tau_2}{M_2\ L_2} + L_1 \dot{\theta}_1{}^2 \sin(\theta_1 - \theta_2) - g\cos(\theta_1) - \cos(\theta_1 - \theta_2)\left[\frac{M\ \tau_1}{M_2\ L_1} + M L_2 \dot{\theta}_2{}^2 \sin(\theta_1 - \theta_2) - g\cos(\theta_1)\right]}{L_2(1 - M\cos^2(\theta_1 - \theta_2))} \tag{32}$$

$$M = \frac{M_2}{M_1 + M_2} \tag{33}$$

Let x_1 represents θ_1, x_2 represents θ_2, x_3 represents $\dot{\theta}_1$ and x_4 represents $\dot{\theta}_2$. The dynamics of the two-link robot arm system are given by following differential equations:

$$\dot{x}_1 = x_3 \tag{34}$$

$$\dot{x}_2 = x_3 \tag{35}$$

$$\dot{x}_3 = g_1(t, x_1, x_2, x_3, x_4, \tau_1, \tau_2) \tag{36}$$

$$\dot{x}_4 = g_2(t, x_1, x_2, x_3, x_4, \tau_1, \tau_2) \tag{37}$$

3. CONTROLLER DESIGN

In this paper, two modified versions of the classical PID controller including the Proportional Integral minus Proportional Derivative (PI-PD) controller and Nonlinear Proportional Derivative (NPD) are used to manipulate the two angular positions of the robot arm and eliminate the effects of the load disturbances to fulfill the application of pick and place tasks of the robot arm. The determination of the design variables of both controllers is essential and it required a suitable cost function to enable the system to reach a stable mode [13] [14]. For this purpose, in Section 4, BA is utilized to find the optimum setting of the adjustable parameters of the controllers.

3.1. PI-PD Controller

The classical Proportional-Integral-Derivative (PID) controller is the most common control strategy that is used in control system design due to its robustness and simplicity [15] [16]. Much research and practice represent a considerable effort to propose a different structure of the classical PID controller. In this direction, this paper utilizes a modified version of the classical PID controller named a Proportional-Integral minus Proportional-Derivative (PI-PD) controller. The general block diagram of the PI-PD controller structure is illustrated in Fig. 2 [17]. The control law (u) of the PI-PD controller is given by [18]:

$$u = K_{p1}e + K_i \int e - K_{p2}y - K_d \frac{dy}{dt} \tag{38}$$

where e and y are the error and the output of the process respectively, K_{p1} and K_{p2} are the proportional gains, K_i is the integrals gain, and K_d is the derivative gain.

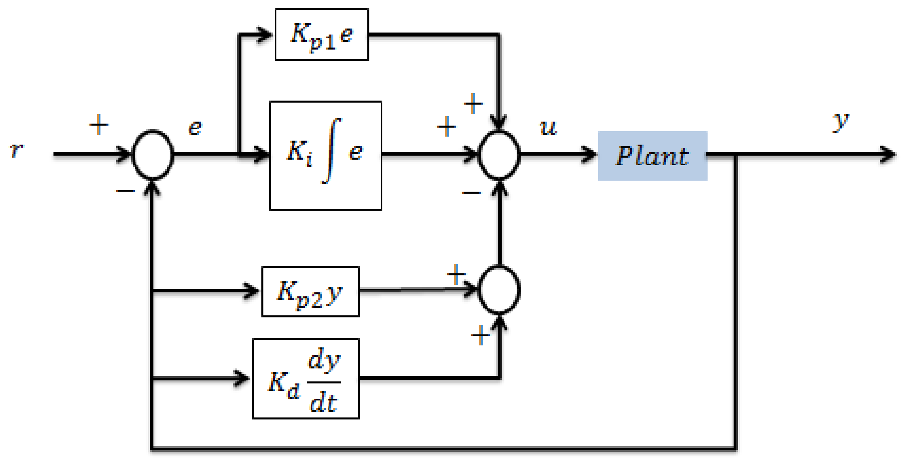

Figure 2. Block diagram of PI-PD controller.

In spite of the simplicity of the PI-PD controller's structure, the tuning process plays a key role in the performance of the PI-PD controller. In order to achieve the optimal performance of the PI-PD controller, BA is employed to find the best value of the design variables of the PI-PD controller.

3.2. Nonlinear PD Controller

To overcome the deficiencies of the classical PID controller for nonlinear systems, a Nonlinear Proportional Derivative (NPD) is introduced. NPD control has been implemented successfully on various process control problems. The control law (u) of the NPD controller is given by [19]:

$$u = K_p f_p(e, \alpha_e, \delta_e) + K_d f_d(\dot{e}, \alpha_{\dot{e}}, \delta_{\dot{e}}) \tag{39}$$

where f_p and f_d are nonlinear function given by [20]:

$$f_p = \begin{cases} |e|^{\alpha_e} sign(e), & for\ |e| > \delta_e \\ \frac{e}{\delta_e^{1-\alpha_e}}, & for\ |e| \le \delta_e \end{cases} \tag{40}$$

$$f_d = \begin{cases} |\dot{e}|^{\alpha_{\dot{e}}} sign(\dot{e}), & for\ |\dot{e}| > \delta_{\dot{e}} \\ \frac{\dot{e}}{\delta_{\dot{e}}^{1-\alpha_{\dot{e}}}}, & for\ |\dot{e}| \le \delta_{\dot{e}} \end{cases} \tag{41}$$

where $\alpha_e, \delta_e, \alpha_{\dot{e}},$ and $\delta_{\dot{e}}$ are additional parameters that describe the behavior of the nonlinear function of the controller. The parameters α_e and $\alpha_{\dot{e}}$ determine the nonlinearity of the nonlinear function. The parameters δ_e and $\delta_{\dot{e}}$ act as a threshold [19]. The Block diagram of the NPD controller is shown in Fig. 3. The selection of the parameter design of the nonlinear PD controller is the major concerns to ensure the stability and to achieve the best performance. To address this issue, BA is introduced to find the optimal value for the adjusted design variables in the NPD controller.

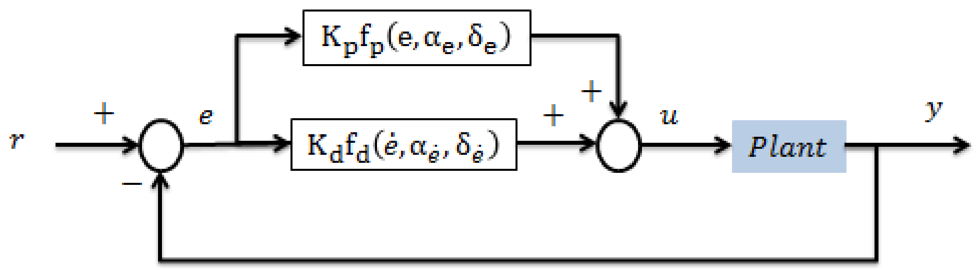

Figure 3. Block diagram of NPD controller.

4. BEE ALGORITHM

Swarm optimization algorithms, in general, are algorithms motivated by the natural behavior of animals, plants, insects… etc. [21]. The Bee Algorithm (BA) is one of the swarm optimization algorithms. The algorithm is introduced by [22]. The algorithm simulates foraging behavior for food of the honey bees. The algorithm starts by initializing the population of bees randomly within the lower and upper boundaries of the search space of the problem as given:

$$p_i = p_{min} + Rand * (p_{max} - p_{min}), \quad i = 1,2,3, \dots, N_{pop} \tag{42}$$

where $i, N_{pop}, and\ p_i$ refer to the index of population, total number of the population, and individual solution respectively, p_{min} and p_{max} are the lower bound and upper bound of the search space, and $Rand$ is a random value between 0 and 1. The BA uses two mechanisms named exploration (i.e. random search) and exploitation (i.e. neighborhood search) to search for the best solution of the optimization problem. In the exploration phase, the algorithm discovers new territory within the search space. On other words, it attempts to find new solutions that haven't been explored previously. In the exploitation phase, the algorithm searches around promising solutions that have showed favorable objective values. In other words, this search converges to the solutions that have already exhibited strong performance. To balance between the exploration search and exploitation search, the BA generates a group of the population with the size of N_m $(i.e. N_m < N_{pop})$ and updates the position of each bee in this group based on the location of the elite bee (p_{ilt}) as given in Eq. (43). The elite bee is the bee that has the best solution that is found by the algorithm.

$$p_i(k+1) = p_i(k) + \sigma(p_i(k) - p_{ilt}) \tag{43}$$

where k is the index of iteration, $p_i(k+1)$ and $p_i(k)$ are the new and current solutions respectively, and σ is the step size. The remaining bees $(N_{pop} - m)$ are assigned to search randomly around the search space as given in Eq. (42). The new population is evaluated and the elite bee is updated. The pseudo code of the BA is given in Algorithm 1.

Algorithm 1 Pseudo code of BA

1. **Input**
   - Objective function, Population size ($\mathrm{N_{pop}}$), Number of iterations ($\mathrm{T_{max}}$), Number of sites ($\mathrm{N_s}$), Step size (σ)
2. **Initialization**
   - Initialize population $\mathrm{N_{pop}}$
   - Evaluate objective function
   - Assign $\mathrm{p_{ilt}}$
3. **Loop:**
   - **while** ($\mathrm{itr} < \mathrm{T_{max}}$)
     - **For** each bee in the sites ($\mathrm{N_s}$)
       - ✓ Update the location of bees using Eq. (3.2)
     - **End for**
     - **For** the remaining bees ($N_{pop} - m$)
       - ✓ Update the location Gorilla using Eq. (4.1)
     - **End for**
     - Update $\mathrm{p_{ilt}}$
     - $\mathrm{itr} = \mathrm{itr} + 1$
   - **End while**
4. **Print the Optimal Solution**

5. SIMULATION RESULTS AND DISCUSSION

To evaluate the performance of the PI-PD controller and NPD controller to control the two-link robot system, the simulation results using MATLAB software are presented in this section. The objective of the controller is to make the two-link in the system follow a desired angular position. The dynamics of the two-link robot arm system, as described by Eq. (34), Eq. (35), Eq. (36) and Eq. (37), are used to conduct the computer simulation. The system's parameters used in the system are listed in Table 1 [1,5].

Table 1. Parameters of two-link robot arm system

Parameters	Values
Mass of first link (M_1)	1 $\mathrm{k_g}$
Mass of second link (M_2)	1 $\mathrm{k_g}$
Length of first link (L_1)	1 m
Length of second link (L_2)	1 m
Acceleration of gravity (g)	9.81 $\mathrm{m/s^2}$

The BA is used for tweaking each controller's design settings in order to guarantee optimal performance. The PI-PD controller is optimized by regulating the adjusted design variables (K_{p1}, K_i, K_{p2} and K_d) of the control action that is introduced in Eq. (38). Similarly, the NPD controller is optimized by regulating the adjusted design variables ($K_p, \alpha_e, \delta_e, K_d, \alpha_{\dot{e}}$ and $\delta_{\dot{e}}$) of the control action that is introduced in Eq. (39). The Integral Time of Absolute Errors (ITAE) index as provided in Eq. (44) [23] is employed as a cost function in the optimization process.

$$ITAE = \int_{tt=0}^{tt=t_{sim}} tt|e(t)|dt \tag{44}$$

where tt is the time and t_{sim} is the total simulation time. Table 2 lists the parameters of the BA. Fig. 4 shows the convergence of BA for tuning the two controllers. Table 3 provides the best values of the designed coefficients of the PI-PD and NPD controllers. Fig. 5 illustrates the time response of the angular angles θ_1 and θ_2 when the system is subjected to a unit step input. The evaluation of the response is performed by measuring the settling time (t_s), the steady state error (e_{ss}), the maximum overshoot, and the ITAE index. These specifications of the two controlled systems are reported in Table 4.

Table 2. Algorithm parameters of BA

Parameters	**Value**
Population Size (N_{pop})	25
Number of Iterations (T_{max})	50
Number of sites (N_s)	15
Step size (σ)	2

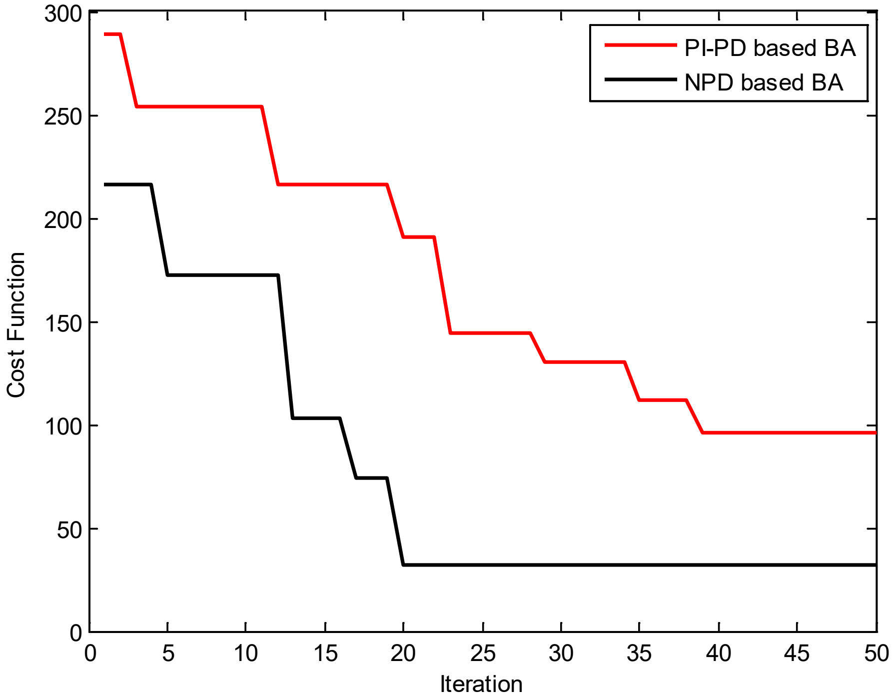

Figure 4. Convergence of BA for proposed controllers

It is evident from Fig. 5 that the two controllers are capable of successfully stabilizing and controlling the system with zero e_{ss}, and zero overshoot response. Regarding t_s and the ITAE index, the dynamics of the NPD controller performs better than the dynamics of the PI-PD controller. For instance, as shown in Table 4, it can be noticed that the value of the t_s is reduced from 2.4 sec and 2 sec for θ_1 and θ_2 response respectively in the case of the PI-PD controller to 1.3 sec and 0.8 sec for θ_1 and θ_2 response respectively in the case of the NPD. This means that the value of t_s is improved by 45.834% and 60% for θ_1 and θ_2 respectively. Furthermore, the value of the ITAE index is reduced from 61.9 and 34.36 for θ_1 and θ_2 response respectively in the case of PI-PD controller to 15.96 and 15.69 for θ_1 and θ_2 response respectively in the case of NPD. This means that the value of the ITAE index is improved by 74.22% and 54.37% for θ_1 and θ_2 respectively. Fig. 6 shows the control signals τ_1 and τ_2.

Table 3. Optimal value of design parameters based on BA

Controller	Parameter	Value
PI-PD	K_{p1}	30
	K_i	20
	K_{p1}	5
	K_d	10
NPD	K_p	60
	α_e	0.2
	δ_e	0.05
	K_d	10
	$\alpha_{\dot{e}}$	0.6
	$\delta_{\dot{e}}$	0.05

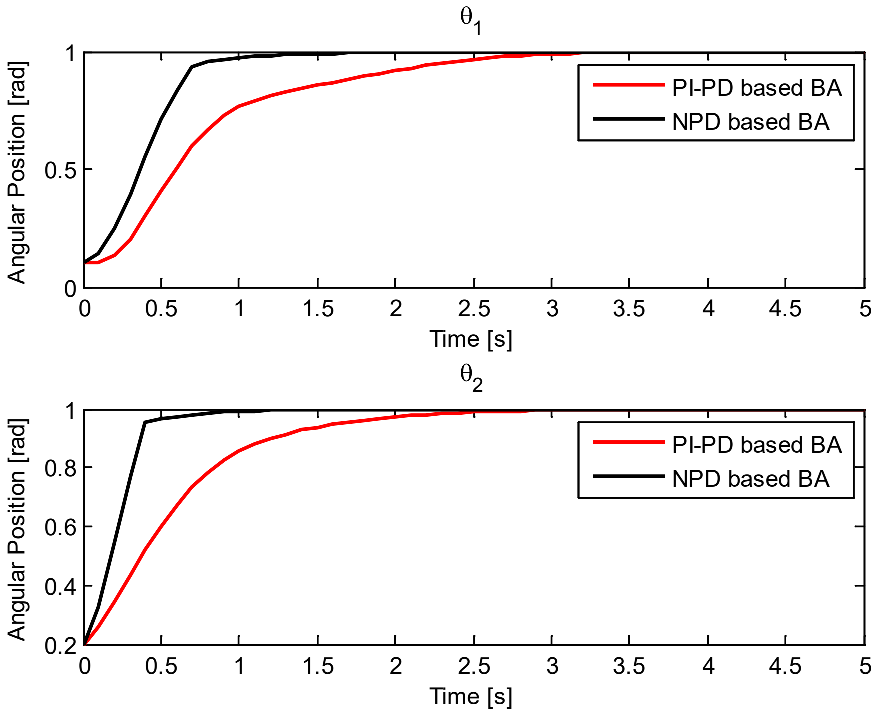

Figure 5. Response of θ_1 and θ_2 for unit step input

Table 4. Specification performances of system without disturbance

Controller	θ	Settling Time (s)	Error Steady State (rad)	Maximum Overshoot (%)	ITAE
PI-PD	θ_1	2.4	0	0	61.9
	θ_2	2	0	0	34.36
NPD	θ_1	1.3	0	0	15.97
	θ_2	0.8	0	0	15.97

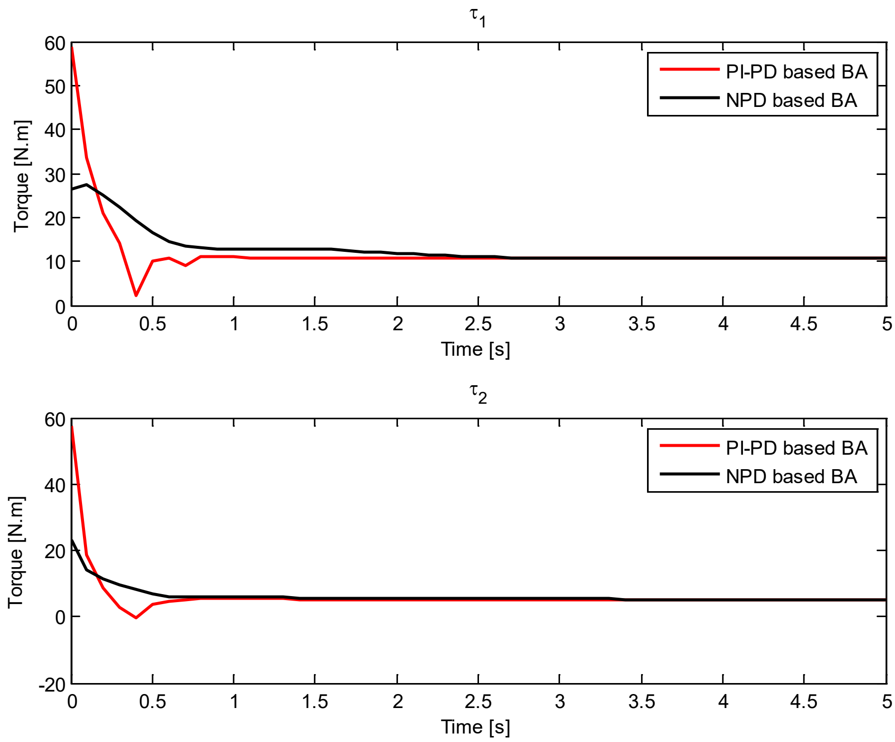

Figure 6. Control signals for unit step input

An external load disturbance has been added to the system based on each controller structure after 4 seconds of simulation time to guarantee the proposed controllers' resilience to load uncertainty. The same designed variables of the controllers that are obtained in Table 3 are used in the simulation. The time response for unit step input is shown in Fig. 7. The system's recovery time and undershoots have been used to evaluate the performance of the system as given in Table 5.

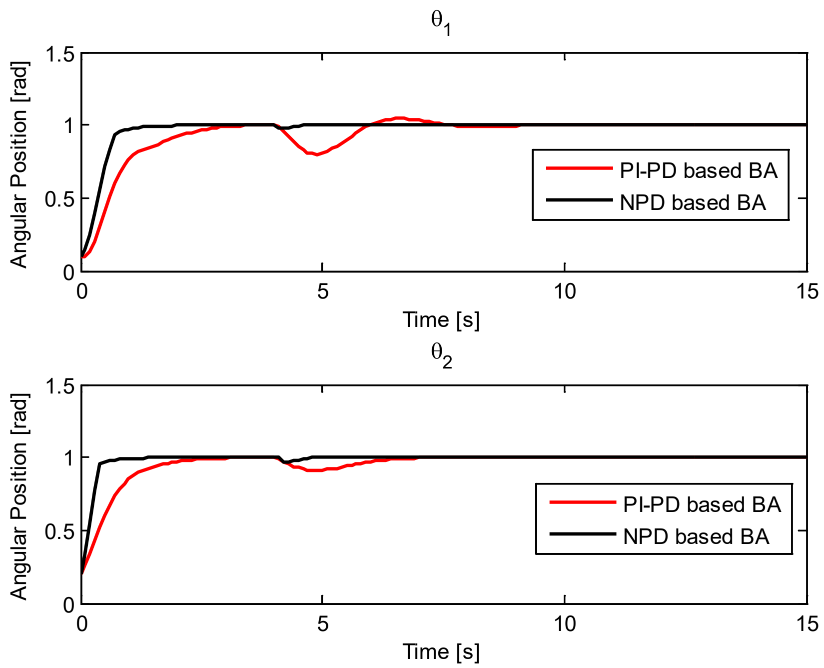

Figure 7. Response of θ_1 and θ_2 for unit step input with disturbance

It is evident from Fig. 7 that in the load disturbance scenario, the NPD controller performs better for recovery form the disturbance than the PI-PD controller. For example, Table 5 shows that the settling time is reduced from 3.3 sec and 2.7 sec for θ_1 and θ_2 response respectively in the case of the PI-PD controller to 0.5 sec and 0.7 sec for θ_1 and θ_2 response respectively in the case of the NPD controller. This means that the value of t_s is improved by 84.5% and 74.1% for θ_1 and θ_2 respectively. Additionally, the maximum undershoot index is reduced from 20% and 9% for θ_1 and θ_2 response respectively in the case of the PI-PD controller to 0.25% and 0.29% for θ_1 and θ_2 response respectively in the case of the NPD. This means that the maximum undershoot is improved by 98.75 and 96.78 for θ_1 and θ_2 respectively. Furthermore, the value of the ITAE index is reduced from 217.96 and 106.87 for θ_1 and θ_2 response respectively in the case of PI-PD controller to 21.188 and 21.188 for θ_1 and θ_2 response respectively in the case of NPD. This means that the value of ITAE index is improved by 89.96% and 80.17% for θ_1 and θ_2 respectively. Fig. 8 shows the control signals τ_1 and τ_2.

Table 5. Specification performances of the system with disturbance

Controller	$\boldsymbol{\theta}$	Recovery Time (s)	Maximum Undershoot (%)	ITAE
PI-PD	θ_1	3.3	20%	217.96
	θ_2	2.7	9%	106.87
NPD	θ_1	0.5	0.25%	21.188
	θ_2	0.7	0.29%	21.188

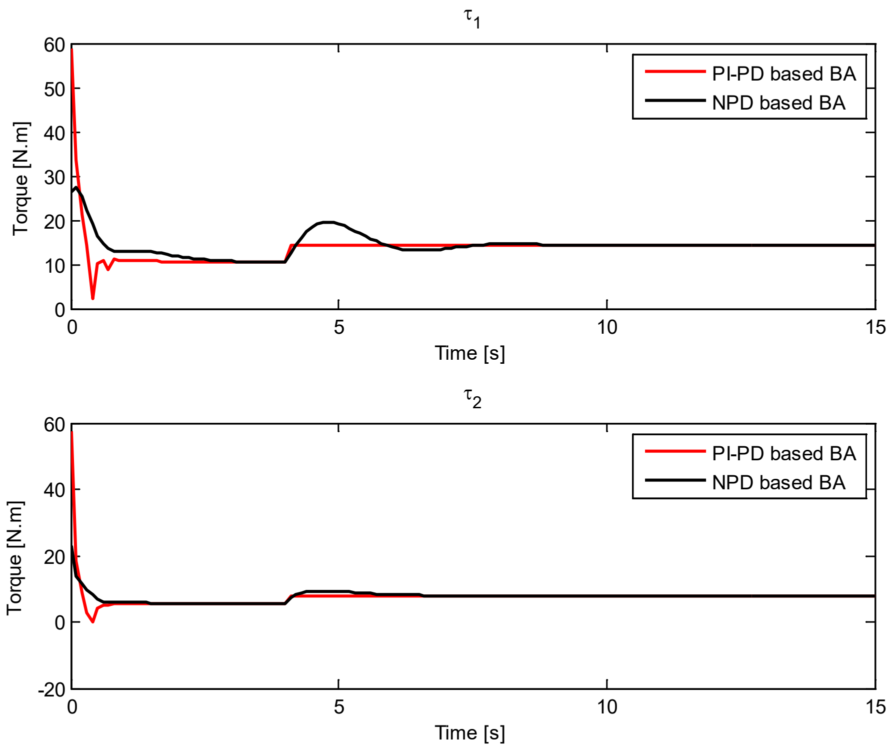

Figure 8. Control signals for unit step input with disturbance

The comprehensive overview of the performance of both controller structures indicates that the NPD outperforms of the PI-PD across the two considered scenarios.

6. CONCLUSION

Controlling the two-link robot arm system was studied based on two control structures named PI-PD and NPD controllers. The dynamics of the system were modeled using Lagrange mechanics. BA was used to adjust the controllers' design parameters in order to guarantee that each controller operated at its optimal performance. According to the simulation results, which were obtained using a MATLAB program, the two controllers optimized by the BA were able to successfully stabilize and regulate the two angular positions of the robot arm system with a 0% error steady state. The outcome also shows that, in terms of reducing the settling time and ITAE index, the NPD-BA controller outperforms the PI-PD-BA controller. Based on the numerical results, the settling time has been improved by 45.834% and 60% for θ_1 and θ_2 respectively whereas the ITAE index has been improved by 74.22% and 54.37% for θ_1 and θ_2 respectively. Furthermore, the NPD-BA controller shows a notable improvement in mitigating the impact of external load disturbances. Based on the numerical results, the settling time has been improved by 84.5% and 74.1% for θ_1 and θ_2 respectively whereas the ITAE index has been improved by 89.96% and 80.17% for θ_1 and θ_2 respectively. This work can be extended further by using another optimization technique to find the adjustment to the controllers' design parameters.

REFERENCES

[1] Baccouch M, Dodds S. (2020) A two-link robot manipulator: Simulation and control design. International Journal of Robotic Engineering, 5(2): 1-17.

[2] Dersarkissian N, Jia R, Feitosa, DL. (2018) Control of a two-link robotic arm using fuzzy logic. In 2018 IEEE International Conference on Information and Automation (ICIA), pp: 481-486.

[3] Velagic J, Hebibovic M, Lacevic B. (2005) On-Line Identification of a Robot Manipulator Using Neural Network with an Adaptive Learning Rate. IFAC Proceedings, 38(1):403-408.

[4] Li-jie C. (2011) Research on the nonlinear dynamical behavior of double pendulum. In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp. 1637-1640.

[5] Mohammed AA, Eltayeb A. (2018) Dynamics and control of a two-link manipulator using PID and sliding mode control. In 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), pp. 1-5.

[6] Guechi EH, Bouzoualegh S. Zennir Y, Blažič S. (2018) MPC control and LQ optimal control of a two-link robot arm: A comparative study. Machines, 6(3): p.37.

[7] Bendimrad A, El Amrani A, El Amrani B. (2022) Optimization of the Performances of a Two-Joint Robotic Arm using Sliding Mode Control. International Journal of Automotive and Mechanical Engineering, 19(2): 9634-9646.

[8] Long Y, You J, Huang Y, Chen Z, Shi M. (2023) Robust Variable Structure PID Controller for Two-Joint Flexible Manipulator. In International Conference on Computing, Control and Industrial Engineering, Singapore: Springer Nature Singapore, pp. 133-144.

[9] Shen Y. (2024) Robotic trajectory tracking control system based on fuzzy neural network. Measurement: Sensors, 31:101006.

[10] Al-Khazraji H. (2022) Comparative study of whale optimization algorithm and flower pollination algorithm to solve workers assignment problem. International Journal of Production Management and Engineering, 10(1):91-98.

[11] Al-Khazraji H, Nasser AR, Khlil S, (2022) An intelligent demand forecasting model using a hybrid of metaheuristic optimization and deep learning algorithm for predicting concrete block production. IAES International Journal of Artificial Intelligence, 11(2): 649.

[12] AL-Khazraji H, Cole C, Guo W. (2021) Optimization and simulation of dynamic performance of production–inventory systems with multivariable controls. Mathematics, 9(5): 568.

[13] Mustafa N, Hashim FH. (2020) Design of a predictive pid controller using particle swarm optimization. International Journal of Electronics and Telecommunications, 66(4): 737-743.

[14] Mahmood ZN, Al-Khazraji H, Mahdi SM. (2023) PID-Based Enhanced Flower Pollination Algorithm Controller for Drilling Process in a Composite Material. In Annales de Chimie Science des Matériaux, 47(2): 91-96.

[15] Mahdi SM, Raafat SM. (2021) Robust Interactive PID Controller Design for Medical Robot System. International Journal of Intelligent Engineering and Systems, 15 (1): 370-382.

[16] Al-Khazraji H. Cole C, Guo W. (2018) Analysing the impact of different classical controller strategies on the dynamics performance of production-inventory systems using state space approach. Journal of Modelling in Management, 13(1): 211-235.

[17] Ozyetkin MM, Onat C, Tan N. (2020) PI-PD controller design for time delay systems via the weighted geometrical center method. Asian Journal of Control, 22(5): 1811-1826.

[18] Peker F, Kaya I, (2017) Identification and real time control of an inverted pendulum using PI-PD controller. In 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), pp. 771-776.

[19] Ouyang PR, Pano V, Tang J, Yue WH. (2018) Position domain nonlinear PD control for contour tracking of robotic manipulator. Robotics and Computer-Integrated Manufacturing, 51:14-24.

[20] Huang H. (2002) Nonlinear PID controller and its applications in power plants. In Proceedings. International Conference on Power System Technology, 3, pp. 1513-1517.

[21] Mahmood ZN, Al-Khazraji H, Mahdi SM. (2023) Adaptive Control and Enhanced Algorithm for Efficient Drilling in Composite Materials. Journal Européen des Systèmes Automatisés, 56(3): 507-512.

[22] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M. (2005) Bee algorithm. Technical Note: MEC 0501. Manufacturing Engineering Centre, Cardiff University, UK

[23] Ahmed AK, Al-Khazraji H. (2023) Optimal control design for propeller pendulum systems using gorilla troops optimization. Journal Européen des Systèmes Automatisés, 56(4): 575-582.