The International Journal of Engineering and Science (IJES)
|| Volume || 12 || Issue || 12 || Pages || PP 39-47 || 2023 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Research on Real-Time Control of Robotic Arm Using Artificial Bee Colony Algorithm

Nguyen Hong Quan, Vu Huy Trung, Nguyen Xuan Thien
Academy of Military Science and Technology, Hanoi, Vietnam
Corresponding Author: quanav08@gmail.com

---*ABSTRACT*---

This article presents the results of controlling a 2-degree-of-freedom (2DOF) robot arm moving in a plane with a traditional PID control system. However, the parameters of this control system are determined using the Algorithm Bee Colony optimization (ABC). The dynamics model of the robot arm is built and serves as the foundation to perform the control problem. The optimal parameters of the PID control system are determined based on the ABC algorithm and through MATLAB/SIMULINK software in real time. The results obtained from the control system are converted into signals that control the actual robot arm model. The results of trajectory control in the joint space show that the PID control system with the support of the ABC algorithm provides high control efficiency with very small errors, the actual joint trajectories closely follow the desired trajectory. This research results are important in confirming the high effectiveness of simple and low-cost traditional control systems when combined with intelligent optimization algorithms. This increases the choice of different control systems for technicians for each specific robot configuration.

KEYWORDS: - *Robot arm, two-link, PID controller, ABC, real time control*

Date of Submission: 01-12-2023 Date of acceptance: 12-12-2023

I. INTRODUCTION

In recent years, robots have become one of the main subjects of research interest because they are applied in many different fields. Indeed, robots are not only applied in industrial production but also in healthcare, education, agriculture and many other areas [1]. Therefore, research on robots is becoming extremely important and urgent. Among the basic problems of robotics, control is always the most important problem because it determines the ability to accurately meet the motion requirements and the level of completion of the initially set task. Accordingly, control algorithms are increasingly developed more powerful and smarter. They start from traditional control systems such as PID [2], Sliding mode [3], to intelligent control systems such as adaptive control [4], fuzzy control [5], neural network-based control [6]. Each type of control system has certain advantages and limitations. Traditional control systems are quite simple in algorithm but their ability to meet accuracy and flexibility in control is not high. In contrast, intelligent control systems allow for improved control accuracy with systems that do not have complete dynamic parameters. They are highly adaptable to such systems.

However, the limitations of these control systems are the complexity of the algorithm and the time cost in setting up and executing the control. Many scientists have aimed to develop more advanced control systems such as PID by applying more intelligent algorithms to search for optimal control parameters such as genetic algorithms [7], swarm algorithm [8], [9], annealing algorithm [10], ant colony algorithm [11], bee colony algorithm [12], or hybrid control systems such as PID-Fuzzy [13], [14], PID-Adaptive [15], PID-NN [16], Adaptive-Fuzzy [17], Fuzzy-NN [18]. However, another important point to mention is the implementation of the above control systems in practice in real time. Building and connecting actual systems is always a necessary issue to be resolved to verify the reliability and stability of control systems designed in theory and simulation.

This problem has been solved by many researchers [19]. The real-time control problem of an adaptive PID controller based on a quantum NN network for mobile robots is presented in [20]. Research results show that the control system responds strongly and effectively in terms of signal processing ability and convergence speed of the algorithm. The real-time hierarchical control system described in [21] aims to meet safety requirements when there is interaction between humans and robots. The controller ensures close monitoring of end operations, singularities and joint limits. The problem of controlling robots in real time with many different time layers is described in detail in [22]. The Zeroing neurodynamics method is presented as one of the effective methods to find equivalence between continuously varying temporal layers during control.

This article describes the control results of the PID-ABC system in real time for a 2DOF robot arm. The dynamic model is built in conjunction with the parameters of the actually manufactured arm system. From the given trajectory, the control parameters of the PID system are searched for optimality by the ABC algorithm. System control results obtained from MATLAB software are transmitted directly to the robot model in reality. The obtained joint angle values are directly converted to motor control signals through the Arduino microprocessor circuit. Actual control results show the feasibility and high efficiency of the PID-ABC system in real time.

II. MATERIALS AND METHODS

2.1. Dynamics modelling

Consider the two-link robot arm model shown in Fig. 1. The robot consists of two-link corresponding to 2DOF. Joint 1 is rotating joint q_1, joint 2 is rotating joint q_2. The joint variable vector is $\mathbf{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$. Point E is the end-effector point with a position vector of $\mathbf{x}_E = \begin{bmatrix} x_E & y_E & z_E \end{bmatrix}^T$. The fixed coordinate system is $(Oxyz)_0$ and is located at the origin O_0. Local coordinate system associated with link 1 $(Oxyz)_1$ is attached at the origin O_1. The end-effector coordinate system $(Oxyz)_E$ is attached at point O_E.

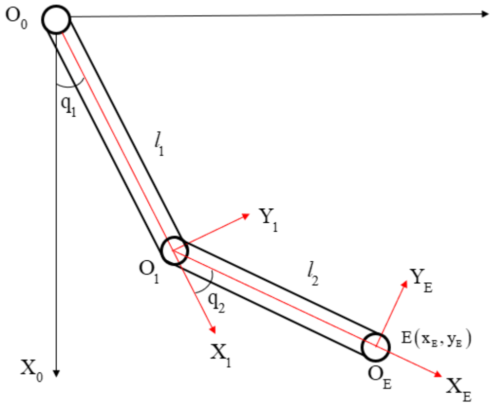


Figure 1. Two-link robot model

The robot configuration parameters according to D-H theory are described in Tab. 1.

Table 1. DH parameters

Links	Parameters			
	θ_i	d_i	a_i	α_i
Link 1	q_1	0	l_1	0
Link 2	q_2	0	l_2	0

The homogeneous transformation matrix of the end-effector link compared to the fixed coordinate system is determined as follows

$$D_2 = \begin{bmatrix} \cos(q_{12}) & -\sin(q_{12}) & 0 & l_1\cos(q_1) + l_2\cos(q_{12}) \\ \sin(q_{12}) & \cos(q_{12}) & 0 & l_1\sin(q_1) + l_2\sin(q_{12}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

In which, $q_{12} = q_1 + q_2$.

The coordinates of the end-effector point are described as follows

$$x_E = l_1\cos(q_1) + l_2\cos(q_{12}); y_E = l_1\sin(q_1) + l_2\sin(q_{12}); z_E = 0 \qquad (2)$$

The dynamics equation is shown as follows

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \qquad (3)$$

In which, $\mathbf{M}, \mathbf{C}$ are the generalized mass matrix and coriolis matrix, respectively. Vectors $\boldsymbol{\tau}$ and $\mathbf{G}$ are the torque vector at the joints and the gravitational potential energy vector.

The mass matrix $\mathbf{M}$ is calculated as follows

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}; M_{11} = m_1 l_{C1}^2 + m_2\left(l_1^2 + l_{C2}^2 + 2l_1 l_{C2}\cos q_2\right) + I_{2z} + I_{1z}$$
$$M_{12} = M_{21} = m_2\left(l_{C2}^2 + l_1 l_{C2}\cos q_2\right) + I_{2z}; M_{22} = m_2 l_{C2}^2 + I_{2z}; l_{C1} = \frac{l_1}{5},\ l_{C2} = \frac{l_2}{2} \tag{4}$$

The Coriolis matrix $\mathbf{C}$ is determined as follows

$$C = \begin{bmatrix} -m_2 l_1 l_2 \sin\left(q_2\right)\dot{q}_2 & -\frac{1}{2}m_2 l_1 l_2 \sin\left(q_2\right)\dot{q}_2 \\ \frac{1}{4}m_2 l_1 l_2 \sin\left(q_2\right)\left(2\dot{q}_1 - \dot{q}_2\right) & \frac{1}{4}m_2 l_1 l_2 \sin\left(q_2\right)\dot{q}_1 \end{bmatrix} \tag{5}$$

In which, m_1 and m_2 are links mass. Moment of inertia of the links are I_{1z} and I_{2z}. Because the robot moves on a horizontal plane, the effects of gravity are balanced and eliminated. Therefore, the potential energy vector $\mathbf{G}\left(\mathbf{q}\right)$ is equal to 0.

2.2. The Artificial Bees Conoly optimization

The Artificial Bees Conoly optimization algorithm based on swarm intelligence was proposed by author Derviş Karaboğa in 2005 [23]. The algorithm gets its idea from the foraging process of bees. Bees live in nests called conoly and leave the nest to search for food (nectar). During that search process, the bees find ways to get the best food source. In the ABC, a beehive will consist of three groups of bees: employed bees, onlookers and scout bees. Assume there is only one worker bee for each food source. In other words, the number of worker bees in the nest will be equal to the number of food sources around. The worker bees will fly to their food source then return to the nest to announce the news. A worker bee whose food source is ignored will become a scout bee and begin searching for a new food source. Observation bees in the nest will rely on information from worker bees to find good food sources. The steps of the algorithm are as follows

Step 1. An initial food source is provided to all worker bees.

Step 2. Repeat steps

Each worker bee flies to the food source according to memory and decides which source is the closest, then evaluates the amount of honey from that source and returns to report.

Each observer bee will observe the worker bee's dance and choose a good food source based on that dance. Next, worker bees will be sent to exploit.

Food sources ignored by observer bees will be replaced by new food sources discovered by scout bees.

The best food sources are saved.

Step 3. The process stops when the request is met

In ABC, the location of food sources represents the solution of the optimization problem and the amount of honey represents the quality of the corresponding solution. The number of worker bees is equal to the number of possible solutions. The ABC algorithm consists of three phases: employ bee phase, onlooker phase and scout bee phase. Each phase will take on a specific role. The entire implementation process of the ABC optimization algorithm is as follows

```
Input: object function, fitness function, lb, ub, n, T, trial, limit
            1. Generate random solutions
            2. Calculate the object function (f) and fitness function (fit) values for the solutions of step 1
            3. Set $trial_h = 0$ for all of the above solutions
for t = 1 to T
            Perform Employed Bees Phase for all solutions
            Calculate the probability of each solution
            Execute the Onlooker Bees Phase to generate a set of solutions
            Save the best solution
      If there is at least 1 solution with trial > limit then
            Perform the Scout Bees Phase with the solution with the largest trial
      end
end
```

In which, T is the number of calculation steps set by the user. The ABC algorithm diagram is shown in Fig. 2.

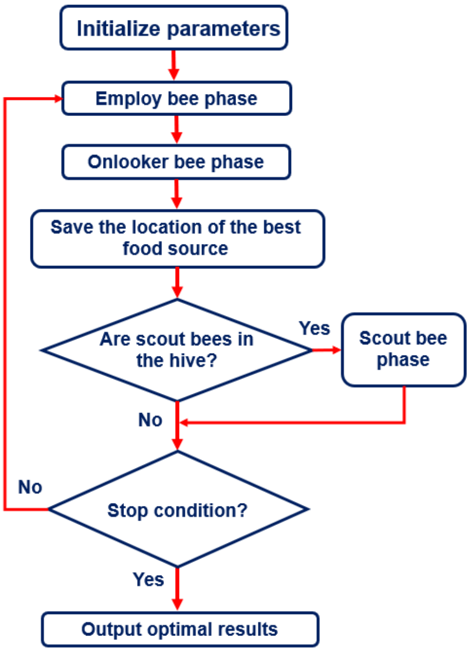
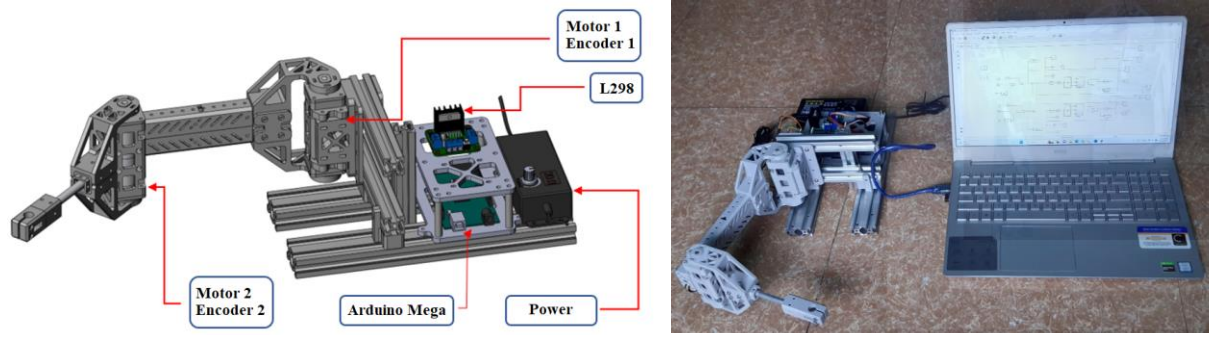


Figure 2. ABC algorithm flow chart

2.3. Control results

Design and fabricate a 2DOF robot arm model located in the horizontal plane as shown in Fig 3.



(a) Design model (b) Fabrication model

Figure 3. 2DOF robot arm model

Geometric and dynamic parameters of the robot arm are described in Tab. 2.

Table 2. 2DOF robot parameters

Parameters	**Link 1**	**Link 2**
Length of links (m) (l_1, l_2)	0,193	0,156
Mass of links (kg) (m_1, m_2)	0,24	0,04
Inertia Momen of links ($kg.m^2$) (I_{z1}, I_{z2})	5.10^{-3}	5.10^{-3}
Joints limits (degree)	-80^o to 80^o	-90^o to 90^o

The given desired joint trajectory in real time is described as follows

$$q_1 = \frac{\pi}{6}\sin(t); q_2 = \frac{\pi}{3}\sin(t) \tag{6}$$

The diagram for implementing the PID control problem with the ABC algorithm is described in Fig. 4.

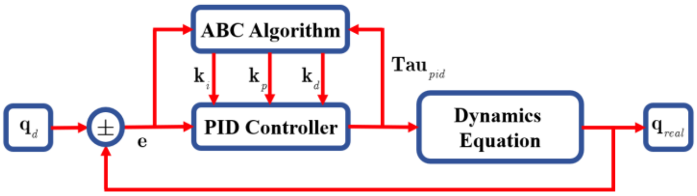


Figure 4. PID-ABC control algorithm diagram

The diagram for connecting devices and transmitting data in real time to control a real robot model is depicted in Fig. 5.

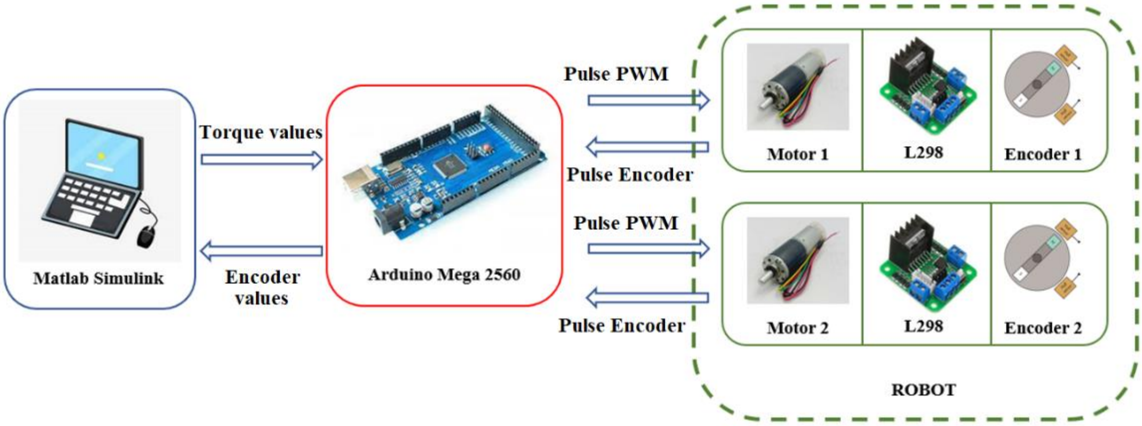


Figure 5. Real-time connection and signal transmission diagram

The robot is set up to connect to the computer. MATLAB/SIMULINK software acts as the central processor and receiving information from the encoder, then performing calculations in the controller and transmitting control signals to the Arduino to control the robot. The Arduino Mega microcontroller acts as a low-level controller to communicate with peripheral devices, including reading pulse signals from the encoder and outputting PWM pulse signals to control the speed and torque of motors. The encoders that come with motors are used to detect the position of the motor shafts as well as the direction of rotation through the number of pulses returned.

The ABC algorithm is used to find 6 parameters including Kp_1, Ki_1, Kd_1 và Kp_2, Ki_2, Kd_2. The parameters of the algorithm are selected as follows: the number of solutions (number of food sources) is n = 5; The number of variables is D = 6; The number of calculation steps is T = 100; The lower boundary value is $\mathbf{lb} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$; The upper boundary value is $\mathbf{ub} = \begin{bmatrix} 100 & 100 & 100 & 100 & 100 & 100 \end{bmatrix}$; The object function is calculated based on the error between the target angle and the actual angle as follows

```
err1 = ref1 - out1
[n, ~] = size(err1)
object_function1 = 0
for i = 1: n
object_function1 = object_function1 + t(i)*abs(err1(i))
end
```

Calculate similarly for the second joint. The values *ref1*, *out1* and *ref2*, *out2* are taken directly while running the simulation using the SIMULINK scheme described in Fig. 6.

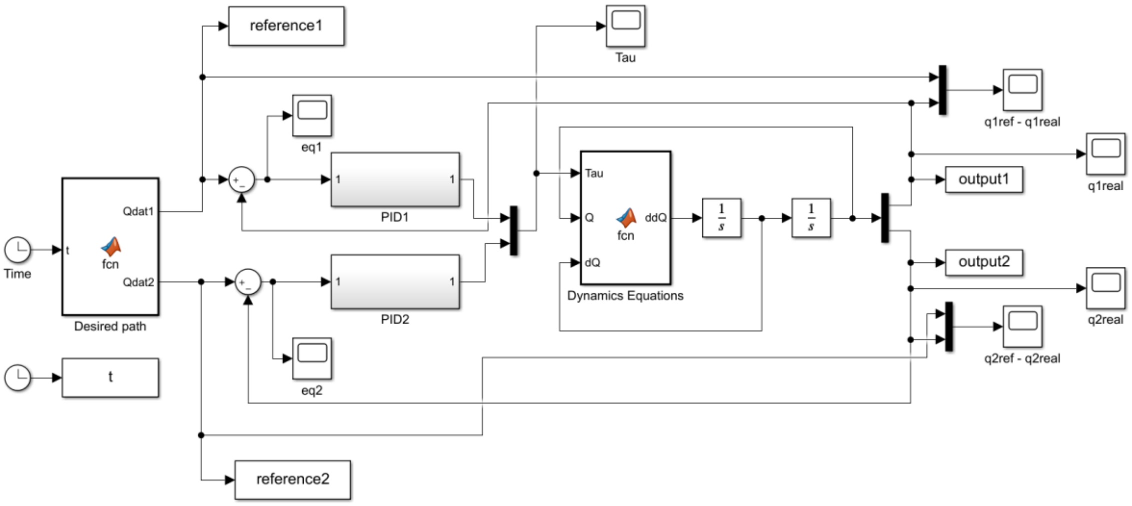


Figure 6. Diagram of calculation and generation of control signals

The objective function value obtained with the ABC algorithm is shown in Fig. 7.

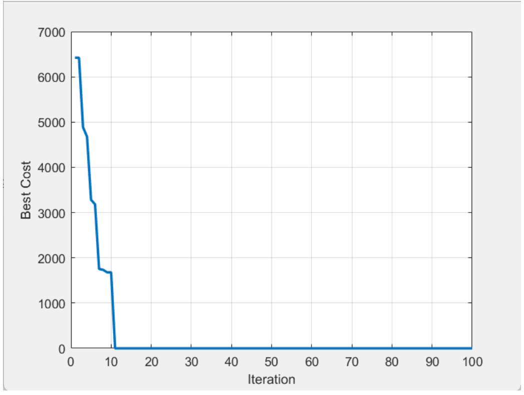


Figure 7. ABC optimization results

The set of values of PID control parameters obtained includes Kp_1 = 79, Ki_1 = 15, Kd_1 = 14 và Kp_2 = 70, Ki_2 = 48, Kd_2 = 0.15. The diagram of connecting the motors to the Arduino Mega processor is described in Fig. 8.

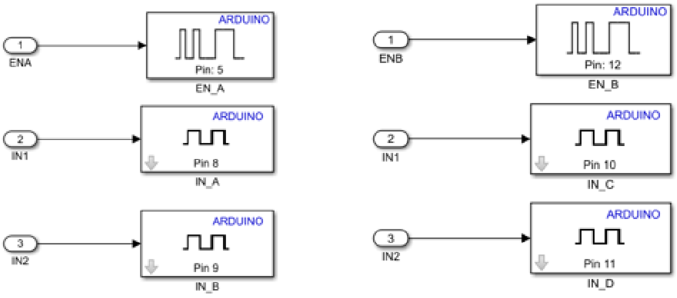


Figure 8. Pinout for Motor 1 and Motor 2

The diagram of transmitting control signals to motor 1 (joint 1) in real time is depicted in Fig. 9. The transmission of control signals to motor 2 is done similarly.

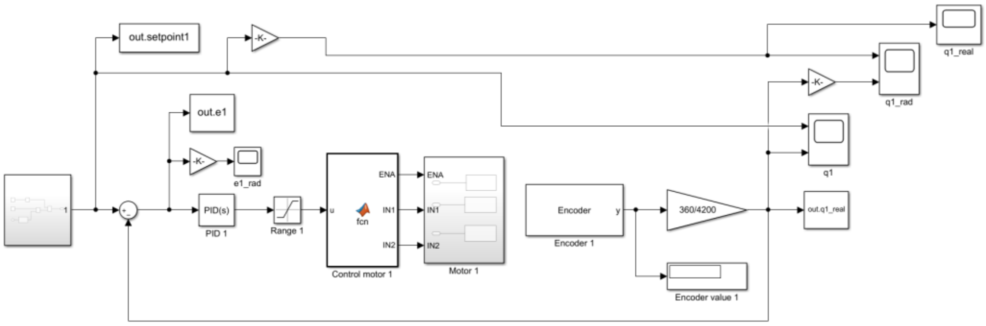


Figure 9. Control signal transmission diagram for drive motor at joint 1

The results of controlling joints 1 and 2 are described in Fig. 10 and Fig. 11. Control errors of joint angles are shown in Fig. 12.

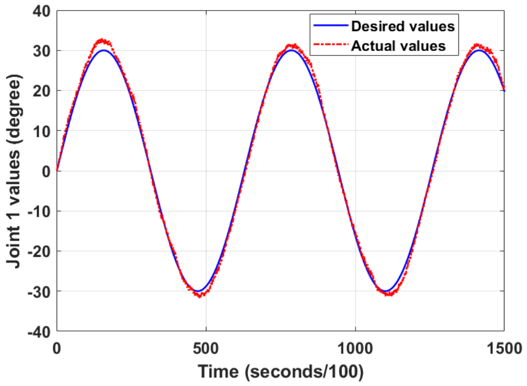


Figure 10. Trajectory control results for joint 1

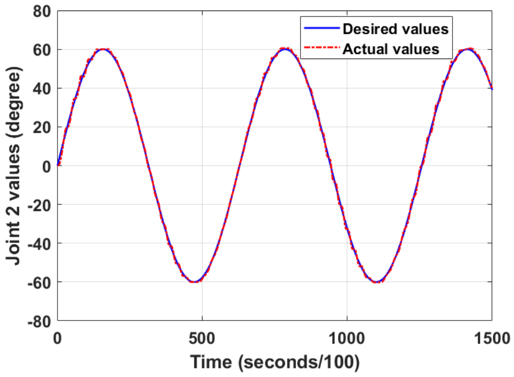


Figure 11. Trajectory control results for joint 2

It is easy to see that joint 1 and joint 2 trajectories are controlled and follow the desired trajectory very well in real time. The error values (in degrees) in both joints are very small. The largest control error value in joint 1 reaches 3.2 (in degrees), and the largest control error value in joint 2 reaches 3.8 (in degrees). So, it can be seen that the PID-ABC control system has responded very well to the problem of controlling the robot arm trajectory in real time with very small errors.

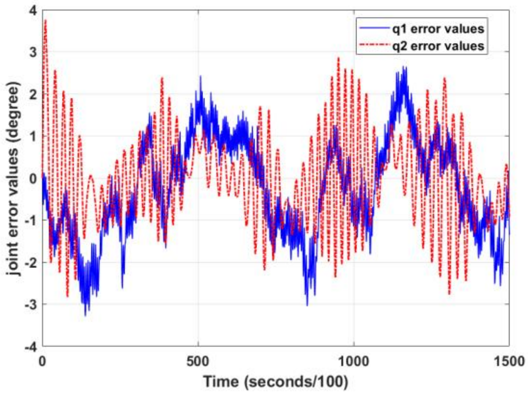


Figure 12. Trajectory control error for joint 1 and joint 2

III. CONCLUSIONS

The control problem a robot arm in real time is a very important and is of interest to research by many scientists. In the process of building a realistic control model, it is necessary to pay attention to many factors such as limiting joint angles to meet the required trajectory completion, and transmitting control data to drive motors from simulate computing environments through microcontrollers and ensure their function properly and efficiently. The above issues were fully addressed in this paper. Accordingly, the efficiency of controlling the 2DOF robot arm in real time is quite high with good trajectory tracking ability and small errors. This research result not only confirms the effectiveness of traditional control systems when combined with optimal algorithms to find a reasonable set of control parameters, but also helps technicians have a basis for choosing suitable control system in real time. This helps reduce time costs and reduce the complexity of the control system while still meeting the initially set goals.

REFERENCES

[1] Kumar, R. Raj, B. Verma, Design of a novel mixed interval type-2 fuzzy logic controller for 2-DOF robot manipulator with payload, Engineering Applications of Artificial Intelligence 123 (2023) 106329.
[2] HVH. Ayala, LS. Coelho, Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator, Expert Systems with Applications 39 (2012) 8968–8974.
[3] N. Zhu, W. Xie, H. Shen, Position-based visual servoing of a 6-RSS parallel robot using adaptive sliding mode control, ISA Transactions xxx (Oct 2023) xxx, https://doi.org/10.1016/j.isatra.2023.10.029
[4] X. Yan, M. Shan, L. Shi, Adaptive and intelligent control of a dual-arm space robot for target manipulation during the post-capture phase, Aerospace Science and Technology 142 (2023) 108688.
[5] T. An, X. Zhu, M. Zhu, B. Ma, B. Dong, Fuzzy logic nonzero-sum game-based distributed approximated optimal control of modular robot manipulators with human-robot collaboration, Neurocomputing 543 (2023) 126276
[6] G. Nadiza, E. Medvet, S. Nichele, S. P. Filho, an experimental comparison of evolved neural network models for controlling simulated modular soft robots, Applied Soft Computing 145 (2023) 110610
[7] M. Vijay, D. Jena, GA Based Adaptive Controller for 2DOF Robot Manipulator, Third International Conference on Advances in Control and Optimization of Dynamical Systems March 13-15, 2014. Kanpur, India.
[8] Z. Bingül, O. Karahan, A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control, Expert Systems with Applications 38 (2011) 1017–1031
[9] F. Liu, W. Liu, H. Luo, Operational stability control of a buried pipeline maintenance robot using an improved PSO-PID controller, Tunnelling and Underground Space Technology 138 (2023) 105178
[10] Z. Wu, J. Dai, B. Jiang, HR. Karimi, Robot path planning based on artificial potential field with deterministic annealing, ISA Transactions 138 (2023) 74–87
[11] L. Wu, X. Huang, J. Cui, C. Liu, W. Xiao, Modifid adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot, Expert Systems with Applications 215 (2023) 119410.
[12] A. Kumar, V. Kumar, Hybridized ABC-GA optimized fractional order fuzzy pre-compensated FOPID control design for 2-DOF robot manipulator, Int. J. Electron. Commun. (AEÜ) 79 (2017) 219–233
[13] V. Kumarn, K.P.S. Rana, Nonlinear adaptive fractional order fuzzy PID control of a 2-link planar rigid manipulator with payload, Journal of the Franklin Institute 354 (2017) 993– 1022
[14] J. Han, X. Shan, H. Liu, J. Xiao, T. Huang, Fuzzy gain scheduling PID control of a hybrid robot based on dynamic characteristics, Mechanism and Machine Theory 184 (2023) 105283
[15] HR. Nohooji, Constrained neural adaptive PID control for robot manipulators, Journal of the Franklin Institute 357 (2020) 3907–3923
[16] J. Perez, J. P. Pereza, et al, Trajectory Tracking Error Using PID Control Law for Two-Link Robot Manipulator via Adaptive Neural Networks, Procedia Technology 3 (2012) 139 – 146
[17] M. H. Barhaghtalab, M. A. Sepestanaki, et al, Design of an adaptive fuzzy-neural inference system-based control approach for robotic manipulators, Applied Soft Computing Journal 149 (2023) 110970
[18] A. F. Amer, E. A. Sallam, W. M. Elawady, Adaptive fuzzy sliding mode control using supervisory fuzzy control for 3 DOF planar robot manipulators, Applied Soft Computing 11 (2011) 4943–4953
[19] J. Li, Y. Zhang, S. Li, M. Mao, New Discretization Formula Based Zeroing Dynamics for Real-Time Tracking Control of Serial and Parallel Manipulators, IEEE Transactions on Industrial Informatics (2017) DOI 10.1109/TII.2017.2780892.
[20] Y. F. Hanna, A. A. Khater, M. El-Bardini, A. M. El-Nagar, Real time adaptive PID controller based on quantum neural network for nonlinear systems, Engineering Applications of Artifiial Intelligence 126 (2023) 106952

[21] Liu, P. Rocco, A. M. Zanchettin, F. Zhao, G. Jiang, X. Mei, A real-time hierarchical control method for safe human–robot coexistence, Robotics and Computer–Integrated Manufacturing 86 (2024) 102666 (in press)
[22] J. Li, X. Zhu, Y. Shi, J. Wang, H. Guo, Real-time robot manipulator tracking control as multilayered time-varying problem, Applied Mathematical Modelling 96 (2021) 355–366
[23] A. Karaman, I. Pacal, et al, Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artifiial bee colony (ABC), Expert Systems with Applications 221 (2023) 119741

Nguyen Hong Quan, et. al. "Research on Real-Time Control of Robotic Arm Using Artificial Bee Colony Algorithm" *The International Journal of Engineering and Science (IJES)*, 12(12), (2023): pp. 39-47.