DE GRUYTER OPEN

1862 RIGA TECHNICAL UNIVERSITY

Electrical, Control and Communication Engineering

ISSN 2255-9159 (online)
ISSN 2255-9140 (print)
2017, vol. 13, pp. 55–62
doi: 10.1515/ecce-2017-0008
https://www.degruyter.com/view/j/ecce

Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm

Nasr A. Elkhateeb* (*Ph.D., Modern University for Technology and Information*),
Ragia I. Badr (*Professor, Cairo University*)

Abstract **– This study presents a well-developed optimization methodology based on the dynamic inertia weight Artificial Bee Colony algorithm (ABC) to design an optimal PID controller for a robotic arm manipulator. The dynamical analysis of robotic arm manipulators investigates a coupling relation between the joint torques applied by the actuators and the position and acceleration of the robot arm. An optimal PID control law is obtained from the proposed (ABC) algorithm and applied to the robotic system. The designed controller optimizes the trajectory of the robot's end effector for a time-variant input and makes the robot robust in the presence of external disturbance.**

Keywords **– Artificial intelligence; Control systems; Evolutionary computation; Robotic manipulators; Trajectory optimization.**

I. Introduction

A robot manipulator is a mobile robot base that includes at least one mounted arm to carry out its functions in an integrated way. The use of mobile manipulators is exponentially increasing in different fields due to factors related to human security. Real-life application environments may be dangerous to human beings or cannot be reached, such as high-temperature sites or ones where harmful gasses are present. The main objective of the manipulator is to reach a certain location and pick up objects. There are two scenarios of using mobile manipulators in industrial fields. The first scenario entails using robot manipulators in transporting and moving objects and tools in known environments. The second one entails using the robots in unstructured environments, especially in dangerous sites that are unsuitable for human beings. Moreover, home assistant robots form another category of autonomous robots that can help mothers at home to perform their daily activities. In all applications, robots are required to move in their environments in order to perform their jobs with a high level of accuracy to achieve reliability.

Control of robot manipulators is a very interesting field due to its complex dynamical model. The dynamical analysis of the robotic model investigates a coupling relation between the joint torques applied by the actuators and the positions of the robotic arm. The non-linear dynamics and the coupling relations make accurate and robust control difficult. So that, designing a controller by the traditional control methods that depend on the robotic system dynamics is a very difficult task. Different control schemes have been provided for the two link manipulator robotic system such as Joel Perez et al. [1]. Perez has introduced a PID control law that depends on neural networks. Also, fuzzy PID controllers have been used in trajectory-tracking robotic systems [2]. Recently, evolutionary algorithms (EAs) have appeared as an alternative design methodology for robotic system applications [3] and [4].

EAs have become an important optimization track for many researchers. These EAs are stochastic optimization methods that imitate natural systems or biologic processes [5]. Being population-based, robustness, and the collective learning process are some of the key features of EAs. The capability to find a global optimum and the possibility to well face nonlinear problems with high numbers of variables are some of the advantages of these algorithms [6]. A bee colony model has been proposed by Karaboga to meet the above requirements. A bee colony is based on the foraging behaviour of honeybee swarms and is employed to solve various optimization problems. A virtual bee algorithm has been presented by Yang [7] to solve the numerical optimization problems. Karaboga [8] has described a bee swarm algorithm called the artificial bee colony (ABC) algorithm.

In the ABC algorithm, solutions are modelled as food sources and their corresponding fitness functions as the quality of the food sources. The ABC algorithm sends artificial bees called employed or onlooker bees to measure the fitness of the food sources (solution candidates). The employed bees are scattered in the problem search domain, producing initial solutions. The numbers of employed and onlooker bees are considered equal until the end of the optimization process. Moreover, like most evolutionary algorithms, the ABC algorithm has some hidden weak points. One of the most important points is the effect of the initial population. The initial population has a negative effect on the overall performance of the algorithm (i.e. the convergence rate and the explorations for the global solution). As pointed out in [9] and [10], the ABC structure supports global exploration more than exploitation. However, the exploration and the exploitation have an essential effect on the performance of the optimization algorithm. More recently, new modifications have been applied to the classical algorithm [11] and [12].

This paper is organized as follows. In Section II, the basic model of the robotic manipulator arm is explained. Also, the main idea in designing the PID control law for the robotic manipulator system and an overview of evolutionary PID controllers are presented. The basic variants of the artificial bee colony are introduced in Section III. The simulation and the results are provided in Section IV. Finally, concluding remarks appear in Section V.

[*] Corresponding author.
E-mail: nasrantar@gmail.com

II. A Dynamic Model of the Robotic System

A robot manipulator is the part of a robot system that is controlled by a human operator and is used to perform specialized tasks in moving and transporting components, tools, etc. The tasks may be specified per a written program to perform different motions [13]. The robot manipulator consists of several links that are connected by joints and move in linear motion. The motion is controlled through several actuators and sensors, which measure the positions of the robot's links.

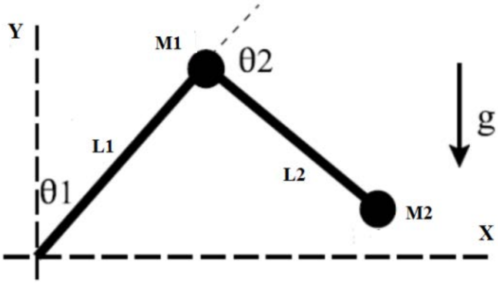


Fig. 1. A robotic model.

Here, M_1 and M_2 are link masses in kg, L_1 and L_2 are the link lengths, g is the gravity acceleration, θ, $\dot{\theta}$, and $\ddot{\theta}$ are the joint positions, velocities, and accelerations, respectively.

The purpose of the control system is to place the end effector and the furthest link at predefined coordinates to perform the task. The actuators move the angles of the joints of the manipulator by applying specific torque.

A. Robotic Arm Dynamics

A simple transformation exists between the Cartesian and polar coordinates of the robot arm segments. For the 2DOF robot manipulator arm, the following direct transformation is used to transform between the XY plane and the angles of the joints.

$$x_1 = L_1\ \sin(\theta_1)\,; \tag{1}$$

$$y_1 = L_1\ \cos(\theta_1)\,; \tag{2}$$

$$x_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)\,; \tag{3}$$

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

The equation that computes the kinetic energy of the above robotic system is described as follows:

$$KE = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2 L_2^2\dot{\theta}_1^2 + \frac{1}{2}M_2 L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2 L_2^2\dot{\theta}_2^2 + M_2 L_1 L_2 \cos\left(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2\right). \tag{5}$$

The potential energy is as follows:

$$PE = M_1 g\, L_1 \cos\theta_1 + M_2 g(L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2). \tag{6}$$

After simplification and transition from Cartesian to polar equations in the kinetic energy equation, the equation that describes the motion of the robot arm manipulator is as follows:

$$B(q)\ddot{q} + C(\dot{q}, q) + g(q) = F, \tag{7}$$

where

$$q = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}. \tag{8}$$

For a robotic manipulator system with n serial links, where $B(q) \in R^{n\times n}$ is a positive definite inertia matrix, $C(\dot{q}, q) \in R^n$ is the vector of centripetal forces, $g(q)$ is the gravity matrix, and $F \in R^n$ denotes the torque applied at the joints. Those matrices describe the dynamic motion of the robotic system:

$$B(q) = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \tag{9}$$

where

$$B_{11} = (M_1 + M_2)L_1^2 + M_2 L_2^2 + 2M_2 L_1 L_2 \cos(\theta_2)\,; \tag{10}$$

$$B_{12} = B_{21} = M_2 L_2^2 + 2M_2 L_1 L_2 \cos(\theta_2)\,; \tag{11}$$

$$B_{22} = M_2 L_2^2; \tag{12}$$

and the inertia and the gravity matrices are

$$C(\dot{q}, q) = \begin{pmatrix} -M_2 L_1 L_2 \sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2 \\ -M_2 L_1 L_2 \sin(\theta_2)(\dot{\theta}_1\dot{\theta}_2) \end{pmatrix}, \tag{13}$$

$$g(q) = \begin{pmatrix} -(M_1 + M_2)gL_1 \sin\theta_1 - M_2 g L_2 \sin(\theta_1 + \theta_2) \\ -M_2 g L_2 \sin(\theta_1 + \theta_2) \end{pmatrix}. \tag{14}$$

The applied torque is

$$F = \begin{pmatrix} f_{\theta_1} \\ f_{\theta_2} \end{pmatrix}. \tag{15}$$

B. Control Implementation

The general structure of the PID controller for any system model is known to consist in the proportional, integral, and derivative actions of the error signal [14]. A mathematical description of the PID controller is as follows:

$$u(t) = K_\mathrm{P}\, e(t) + K_\mathrm{I} \int e(t)dt + K_\mathrm{D}\frac{de(t)}{dt}. \tag{16}$$

For the robot manipulator model, the controller output is the torque applied to the robot dynamic system as follows:

$$\ddot{q} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + F, \tag{17}$$

where $\hat{F}$ is the non-physical torque and related to the actual input torque through inertia matrix $B(q)$ as follows:

$$\hat{F} = B(q)^{-1}F \ \leftrightarrow\ F = B(q)\hat{F}.$$

By decoupling the above system to have the non-physical input torque, the following is obtained:

$$\hat{F} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{18}$$

The physical input torque of the dynamic system is as follows:

$$\begin{pmatrix} f_{\theta_1} \\ f_{\theta_2} \end{pmatrix} = B(q)\begin{pmatrix} f_1 \\ f_2 \end{pmatrix}; \tag{19}$$

$$f_1 = K_{\mathrm{P}1}\left(\theta_{1f} - \theta_1\right) + K_{\mathrm{I}1} \int e(\theta_1)dt - K_{\mathrm{I}1}\dot{\theta}_1; \tag{20}$$

$$f_2 = K_{\mathrm{P}2}\left(\theta_{2f} - \theta_2\right) + K_{\mathrm{I}2} \int e(\theta_2)dt - K_{\mathrm{I}2}\dot{\theta}_2. \tag{21}$$

So, the complete system equations are as follows:

$$\ddot{q} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + \begin{pmatrix} K_{\mathrm{P}1}(\theta_{1f} - \theta_1) + K_{\mathrm{I}1} \int e(\theta_1)dt - K_{\mathrm{I}1}\dot{\theta}_1 \\ K_{\mathrm{P}2}(\theta_{2f} - \theta_2) + K_{\mathrm{I}2} \int e(\theta_2)dt - K_{\mathrm{I}2}\dot{\theta}_2 \end{pmatrix}. \quad (22)$$

The goal now is to find the proper coefficients $K_{\mathrm{P}}, K_{\mathrm{I}},$ and K_{D} for each joint in order to minimize the trajectory errors of the joints (22).

C. Cost Function

The most vital point in achieving a high-fitness optimization process is to select an objective function which is used in evaluating the fitness of the candidate solution. Good formulation of the fitness function leads to optimal solutions during the optimization process. In this robotic system, three different objective functions are chosen in the control parameter optimization process (23)–(25).

Mean of Root of Squared Error (MRSE)

$$MRSE = \frac{1}{N}\sum_{i=0}^{N} \sqrt{e_{\theta_1}(i)^2 + e_{\theta_2}(i)^2}. \quad (23)$$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{N}\sum_{i=0}^{N} |e_{\theta_1}(i)| + |e_{\theta_2}(i))|. \quad (24)$$

Reference Based Error with Control Effort (RBECE)

$$RBECE = \frac{1}{N}\sum_{i=0}^{N} |e_{\theta_1}(i)| + |e_{\theta_2}(i)| + |u_{\theta_1}(i)| + |u_{\theta_2}(i)|. \quad (25)$$

These objective functions are calculated based on the dynamically response of the robotic system [16].

III. Artificial Bee Colony (ABC)

The artificial bee colony (ABC) algorithm is an evolutionary algorithm derived from the foraging behaviour of the honeybee. Honeybee swarms consist of three essential components: employed foragers, unemployed foragers, and food sources. It defines two leading modes of the honeybee colony behaviour: recruitment to a food source and abandonment of a source [8]. In an ABC algorithm, the artificial colony consists of three types of bees: employed, onlookers, and scout bees. At the same time, half of the artificial colony consists of employed bees, and the other half includes the onlooker bees. To guarantee the diversity of the solution space and prevent the algorithm from being trapped in local optimum, scout bees are added in each generation. Each food source has only one employed bee. So, the number of food sources is equal to the number of employed bees inside the colony. In every round, food sources are scattered with their neighbours to produce new solutions and then evaluated based on the fitness function. A solution candidate that does not produce improvement in solutions is assumed to be an abandoned source and is replaced with a new solution [15].

A. Employed Bee Phase

In this phase, ABC sends employed bees onto the food sources and then measures the source quality (i.e. quantity, richness, proximity, etc.). The employed bees carry this information to the hive and then on the dancing area (area of information exchange) share it with the onlookers. The employed bee memorizes its food source and then either continues at the same food source or selects a new one. An artificial bee produces a new solution with the following formula.

$$X_{i,j}(t+1) = x_{i,j} + \Phi_{ij}\left(x_{k,j}(t) - x_{i,j}(t)\right). \quad (26)$$

$X_{i,j}$ is a new solution that comes from food source $x_{i,j}$ and its neighbour $x_{k,j}$, where j and k are two random indices, Φ is a randomly produced number in the range [–1; 1], and t is the generation time.

Moreover, the performance of the bee colony can be improved by controlling the impact of the initial population on the new produced food sources. It was shown in [7] that the introduction of a dynamic inertia weight parameter in the basic (ABC) equation (26) can play a positive role in controlling the impact of the previous solution on the new expected one as follows:

$$X_{i,j}(t+1) = \omega_i x_{ij}(t) + \Phi_{ij}\left(x_{k,j}(t) - x_{i,j}(t)\right). \quad (27)$$

The inertia weight $\omega_i \in [0;\ 1]$ is employed to manipulate the impact of the previous history of velocities on the current velocity. Therefore, ω_i resolves the tradeoffs between the global (wide-ranging) and local (nearby) exploration ability of the swarm [17].

$$\omega_i = (\omega_{\mathrm{init}} - \omega_{\mathrm{fin}})\left(\frac{MaxCycles - i}{MaxCycles}\right) + \omega_{\mathrm{fin}}, \quad (28)$$

where ω_{init} is the initial inertia weight, ω_{fin} is the final inertia weight, $MaxCycles$ is the maximum iteration value and i is the variable iteration index. Note here that the inertia weight ω_i plays an important role in the convergence of the ABC algorithm to the global optimal solution and hence has an influence on the time taken for a simulation run.

A large inertia weight encourages global exploration (moving to previously encountered areas of the search space), while a small one promotes local exploration, i.e. fine-tuning the current search area. A suitable value for ω_i provides the desired balance between the global and local exploration ability of the swarm and, consequently, improves the effectiveness of the algorithm [17].

B. Onlooker Bee Phase

The onlookers decide and select the food source depending on the nectar information (i.e. quality, amount, distance between the food source and the hive, etc.). The probability of a certain food source being selected increases as the information received from the dancing area means that a large amount of high-quality nectar exists there. The onlookers choose a food source with a probability calculated by using various schemes. In this study, an artificial onlooker bee chooses a food source with probability p_i expressed as follows:

$$p_i = a\frac{fit_i}{\sum_i^k fit} + b, \quad (29)$$

where a and b are two arbitrary numbers in range [0; 1], fit_i is the fitness value of solution i, and k is the number of employed bees. Basturk [15] has expressed the fitness function as

$$fit_i = \begin{Bmatrix} \frac{1}{1+F_i} & \text{if } F_i \geq 0 \\ 1 + \text{abs}(F_i) & \text{if } F_i < 0 \end{Bmatrix}, , \qquad (30)$$

where F_i is the objective function to be optimized.

C. Scout Phase

When an employed bee decides to leave its food source, it becomes a scout. In ABC, if the position of a food source cannot be improved further through a predetermined number of cycles, then that food source is discarded. The value of the predetermined number of cycles is an important control parameter of the ABC algorithm, which is called the limit for abandonment. The scout phase plays a positive role in adding newcomers to the solution space (i.e. by generating new random solutions instead of abandoned food sources); however, the number of scouts is limited by the colony size and the dimensions of the problem.

IV. Simulations and Results

Numerical results of computer simulations have been obtained to assess the capabilities of the proposed tuning procedure.

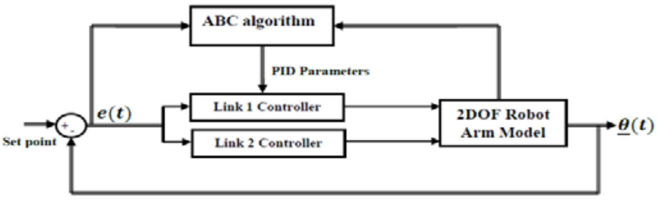


Fig. 2. Robotic system PID tuning schematic diagram.

For the robotic system, the masses of the two links are $M_1 = 1$ kg and $M_1 = 1$ kg, the lengths are $L_1 = 1$ m and $L_2 = 1$ m. The gravity acceleration $g = 9.81\ \text{m/s}^2$ and the simulation period is taken to be 20 s.

The proposed robotic arm trajectory is considered as a sinusoidal signal [16]:

$$\theta_{1f}(t) = 0.1524 + 0.24384 \cos\left(\frac{2\pi t}{5} - \frac{\pi}{2}\right); \qquad (31)$$

$$\theta_{2f}(t) = 0.39624 + 0.24384 \cos\left(\frac{2\pi t}{5} - \frac{\pi}{2}\right). \qquad (32)$$

For the ABC algorithm, the size of the colony is 20 (employed bees + onlooker bees), the number of scouts is equal to one, the number of food sources is equal to half the colony size, i.e. 10. The bounds for the PID parameters are [0; 100]. The constants of (29) are chosen to be $a = 0.9$, and $b = 0.1$. The dynamic range of the inertia weight is [0.6; 1], where $\omega_{\text{init}} = 0.6$, and $\omega_{\text{fin}} = 1$. Finally, the maximum number of cycles is equal to 100.

Several simulation experiments have been carried out to design an optimal PID controller for a 2DOF Robotic manipulator using the proposed dynamic inertia weight artificial bee colony algorithm. The internal structure of the robotic system is built by using Simulink MATLAB Software Tool (Fig. 3). The optimization process is carried out to minimize the error in the trajectory of the robotic manipulator's end effector.

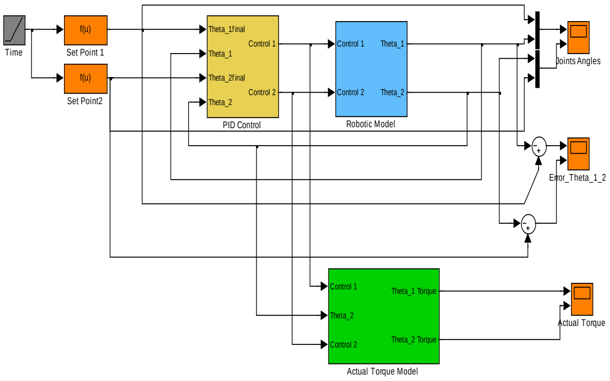


Fig. 3. The Simulink model for the robotic system.

TABLE I
Joint-1/2 Parameters for the Three Cost Functions

Index	*MRSE*	*MAE*	*RBECE*
K_{P1}	200.0000	200.0000	20.4133
K_{I1}	22.3749	50.0000	2.0000
K_{D1}	23.7962	6.9577	28.0853
K_{P1}	184.0705	189.496	41.7381
K_{I1}	50.0000	47.2071	4.9652
K_{D1}	7.9692	7.2797	3.8269
Cost Function	0.0011452	0.04636	12.1983

Table I provides the optimal controller parameters and the corresponding values of objective functions with three different objective functions.

The simulation results and the performance of the robotic system during the optimization of the MRSE function are given. The actual positions of the joints and the desired set point are shown in Fig. 4, the resultant errors in Fig. 5 and the actual torques in Fig. 6. Fig. 7 shows the trajectory of the robotic manipulator's end-effector in the *XY* plane.

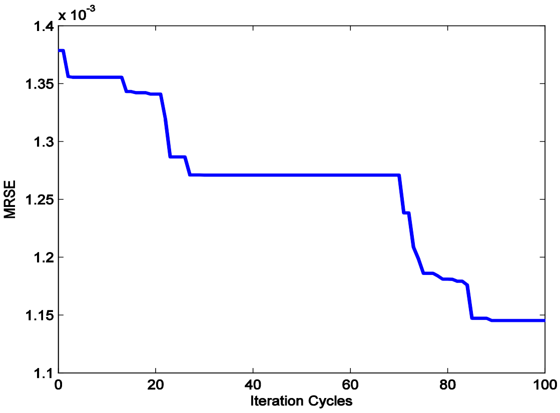


Fig. 4. Convergence history of MRSE function.

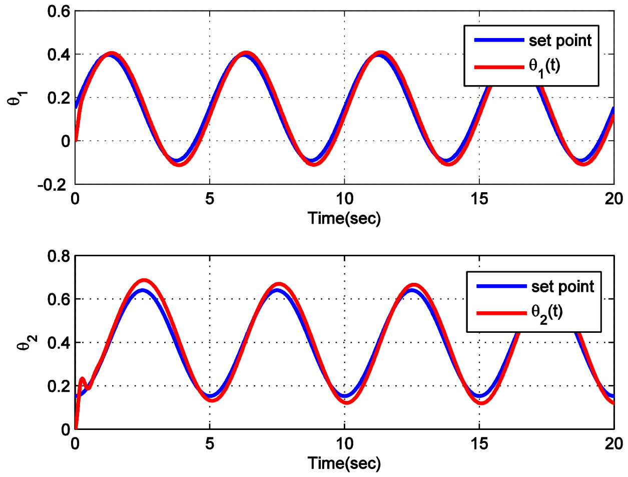


Fig. 5. MRSE joints trajectories.

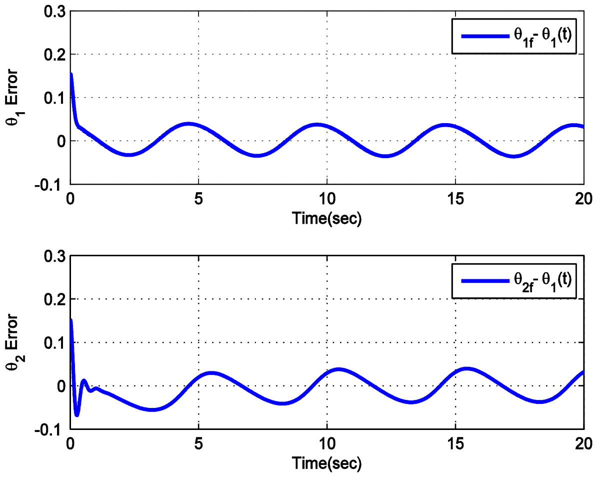


Fig. 6. MRSE tracking errors.

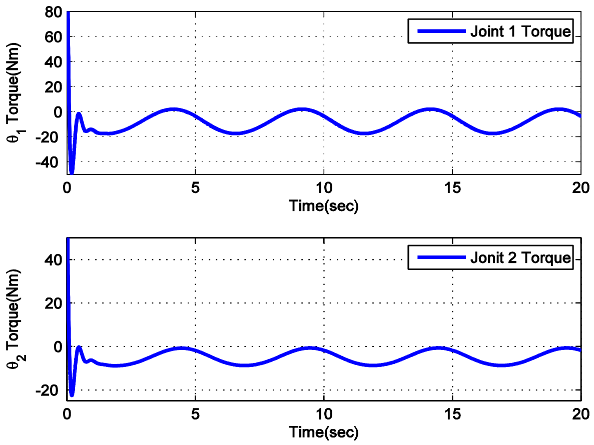


Fig. 7. MRSE optimal applied torque.

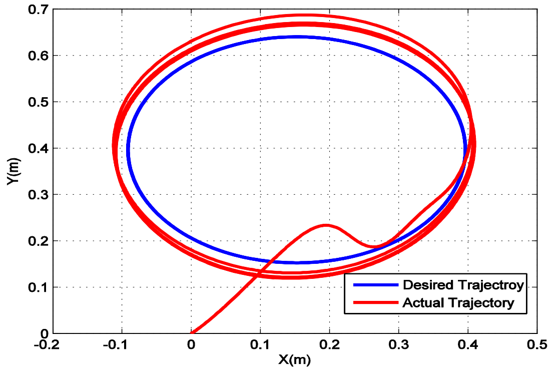


Fig. 8. MRSE robotic arm trajectory in *XY* plane.

Figures 9–13 give the results of the robotic system during the optimization of the MAR function. The actual positions of the joints and the desired set point are given in Fig. 10, the resultant errors in Fig. 11 and the actual torques in Fig. 12.

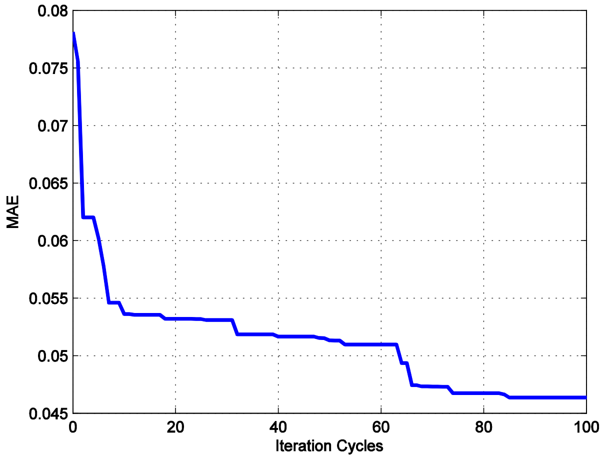


Fig. 9. Convergence history of MAE function.

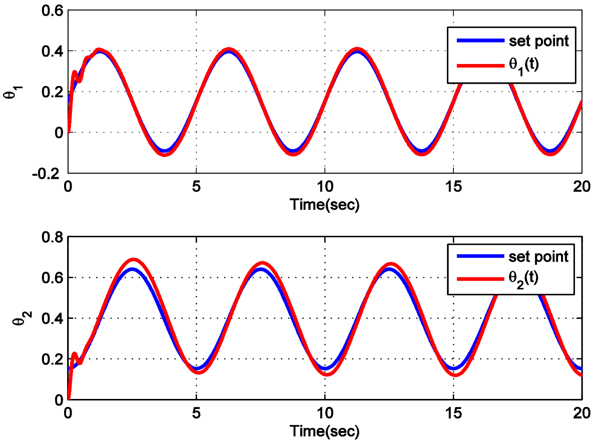


Fig. 10. MAE joints trajectories.

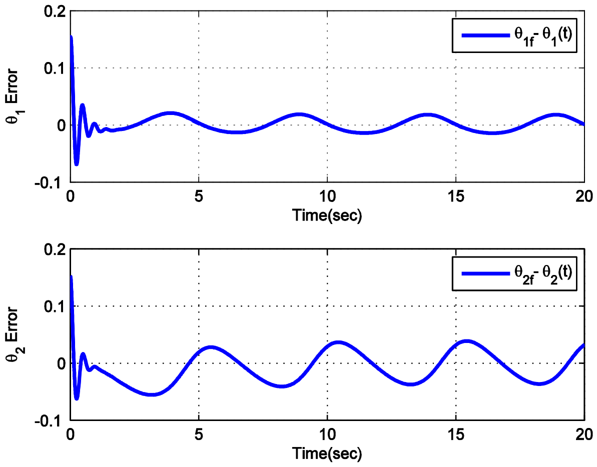


Fig. 11. MAE tracking errors.

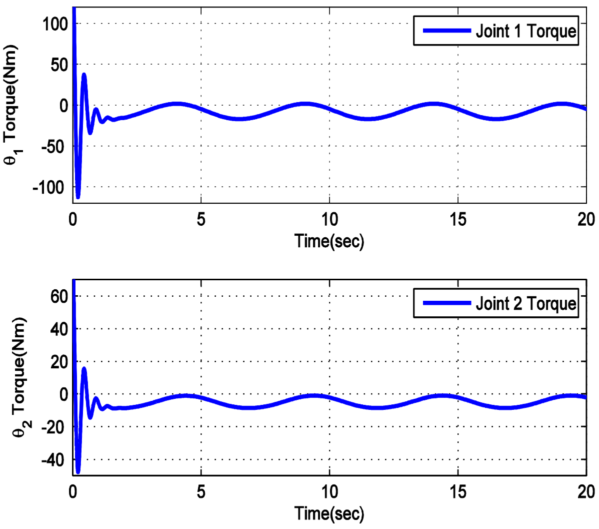


Fig. 12. MAE optimal applied torque.

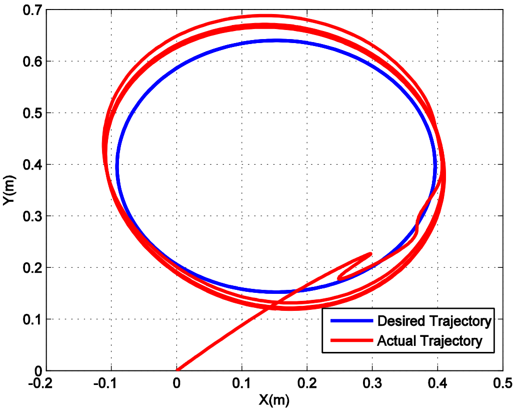


Fig. 13. MAE robotic arm trajectory in *XY* plane.

The simulation results for the robotic system during the optimization of the RBECE function are given in Fig. 14–Fig. 17. The actual joints positions and desired set point are given in Fig. 15, the resultant errors in Fig. 16 and the actual torques are shown in Fig. 17.

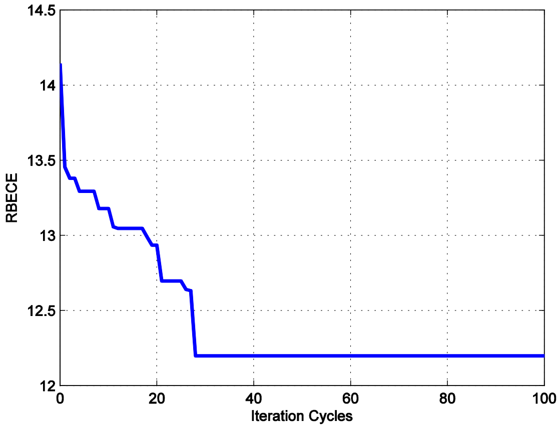


Fig. 14. Convergence history of RBECE function.

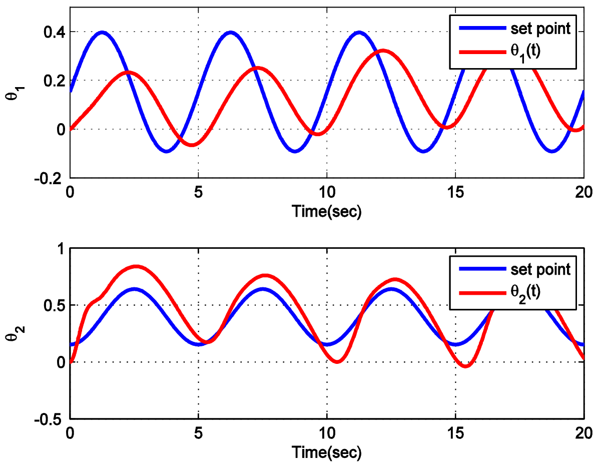


Fig. 15. RBECE joints trajectories.

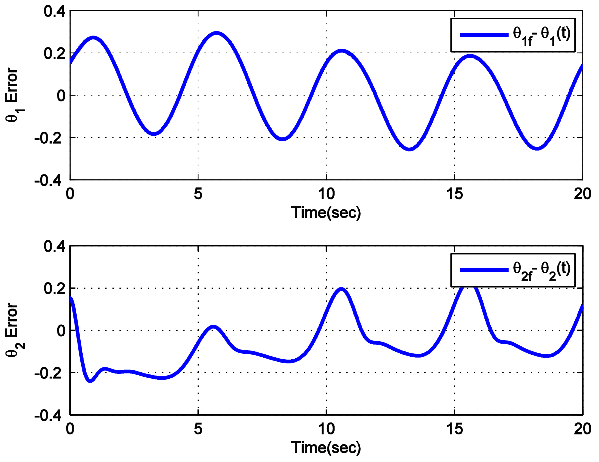


Fig. 16. RBECE tracking errors.

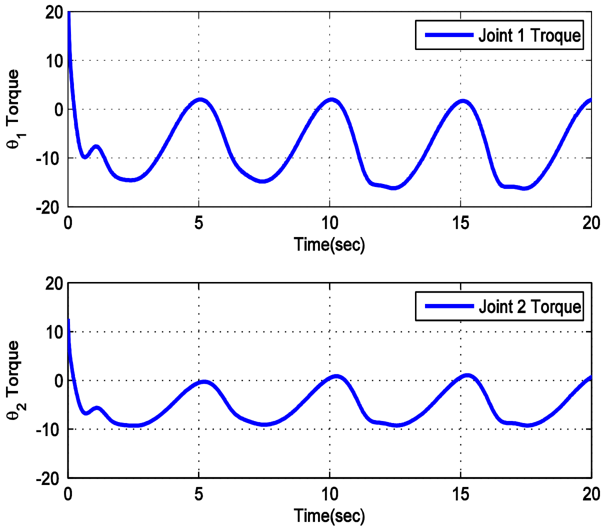


Fig. 17. RBECE optimal applied torque.

The convergence history of the three objective functions MRSE, MAE, and RBECE is shown in Fig. 4, Fig. 9 and Fig. 14, respectively. As can be seen, the objective functions converge to the optimal solution within a shorter generation time. For the RBECE function, the optimization process reaches the optimal solution in one quarter of the generation cycles.

A. Robustness Analysis

The objective of this section is to test the robustness of the optimization algorithm results when the robotic system is subjected to external noise power. To check the robustness of the obtained controller, an external disturbance is added to the output interface between the applied controller torque and the robotic system, as can be seen in Fig. 18. The external disturbance is implemented with a white noise with different variances.

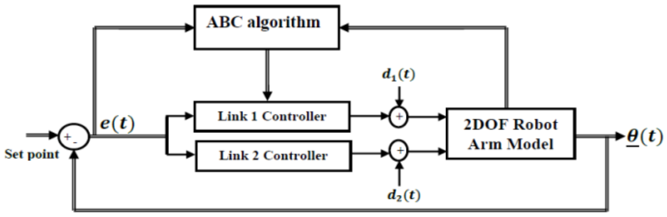


Fig. 18. Schematic of robotic system tuning in the presence of noise.

The objective functions MRSE, MAE and their corresponding PID parameters are shown in Tables II and III at different values of the noise power.

TABLE II

MRSE Cost Function Values in the Presence of Noise Power

σ	K_{P1}	K_{I1}	K_{D1}	K_{P2}	K_{I2}	K_{D2}	*MRSE*
0.0	1000	11.9	21	1000	724.9	263	0.001801
0.1	879.3	705.1	14.6	1000	1000	202.2	0.001117
0.2	700.4	57.16	20	809	1000	185.1	0.001068
0.5	772.1	88.7	14.4	678.8	1000	140.8	0.000972
0.7	818.6	169	21	954.1	1000	156.9	0.000893
0.8	923.2	153.6	18.6	1000	1000	184.8	0.000852

TABLE III

MAE Cost Function Values in the Presence of Noise Power

σ	K_{P1}	K_{I1}	K_{D1}	K_{P2}	K_{I2}	K_{D2}	*MAE*
0.0	801.3	1000	21	397.7	1000	51.2	0.0499688
0.1	937.3	597.9	19.3	463.2	1000	50.2	0.0363772
0.2	1000	696.1	19.8	687.9	850.9	52.4	0.0347977
0.5	893.7	1000	36.4	1000	1000	18.9	0.0322880
0.7	1000	804.6	22.7	627.1	1000	40.2	0.0302509
0.8	1000	701.9	17.1	496.8	938.4	36.8	0.0306091

As can be seen from the robustness results in Tables II–III, the disturbance has an effect on the PID parameters and the corresponding cost functions. When the noise power increases, a small deviation from the desired set point trajectory occurs but the overall arm performance is acceptable.

B. Remark

From the observations of the simulations and the results, the following can be concluded:

i. The proportional gain K_P is directly related to the error and speed of the joints.

ii. The differential gain K_D is directly related to the speed of interaction with state changes'

iii. The integral gain K_I is directly related to the overall joint error cancellations

The above arguments are rough due to the high order of nonlinearity and the coupling equations of the robotic system. The dynamic system produces interactions between different controller components, which lead to small changes in controller parameters and unexpected performance. Moreover, the controller parameters are highly coupled with the initial and final conditions of the link joints. On-line tuning of the controller parameters is highly necessary to cover global system operation.

V. Conclusion

A novel tuning methodology based on the artificial bee colony algorithm has been presented for trajectory tracking in robotic arm manipulator systems. The proposed PID control law is obtained and the optimal gains are tuned by using a dynamic inertia weight artificial bee colony optimization algorithm. The obtained results are satisfactory and competitive; where the robotic system is highly nonlinear, serious system difficulties have to be tuned by traditional methods. To ensure the robustness of the obtained control law, a robustness test is carried out for the robotic system in the presence of external disturbance input. Furthermore, the tuning of the PID parameters with the ABC algorithm is easier than traditional methods that require derivatives and complex mathematical solutions.

References

[1] J. P. P., J. P. Perez, R. Soto, A. Flores, F. Rodriguez, and J. L. Meza, “Trajectory Tracking Error Using PID Control Law for Two-Link Robot Manipulator via Adaptive Neural Networks,” *Procedia Technology*, vol. 3, pp. 139–146, 2012. https://doi.org/10.1016/j.protcy.2012.03.015

[2] S. A. Ahmed and M. G. Petrov, “Trajectory Control of Mobile Robots Using Type-2 Fuzzy-Neural PID Controller,” *IFAC-PapersOnLine*, vol. 48, no. 24, pp. 138–143, 2015. https://doi.org/10.1016/j.ifacol.2015.12.071

[3] P. V. Savsani and R. L. Jhala, “Optimal Motion Planning For a Robot Arm by Using Artificial Bee Colony (ABC) Algorithm,” *International Journal of Modern Engineering Research (IJMER)*, vol. 2, no. 6, pp. 4434–4438, 2012.

[4] Q. Ma and X. Lei, “Dynamic Path Planning of Mobile Robots Based on ABC Algorithm,” *Lecture Notes in Computer Science*, pp. 267–274, 2010. https://doi.org/10.1007/978-3-642-16527-6_34

[5] T. Back, *Evolutionary Algorithms in Theory and Practice*. London, UK: Oxford University Press, 1996. http://doi.org/10.1002/(SICI)1099-0526(199703/04)2:4<26::AID-CPLX6>3.0.CO;2-7

[6] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg, 2008. https://doi.org/10.1007/978-3-540-73190-0

[7] X.-S. Yang, “Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms,” *Lecture Notes in Computer Science*, pp. 317–323, 2005. https://doi.org/10.1007/11499305_33

[8] D. Karaboga, “An Idea Based on Honey Bee Swarm for Numerical Optimization,” *Lecture Notes in Computer Science Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[9] N. Elkhateeb and R. Badr, “Employing Artificial Bee Colony With Dynamic Inertia Weight for Optimal Tuning of PID Controller,” *5th International Conference on Modeling, Identification and Control*, pp. 42–46, Cairo, Egypt, 2013.

[10] G. Zhu and S. Kwong, “Gbest-Guided Artificial Bee Colony Algorithm for Numerical Function Optimization,” *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, Dec. 2010. https://doi.org/10.1016/j.amc.2010.08.049

[11] N. Elkhateeb and R. Badr, “A Novel Variable Population Size Artificial Bee Colony Algorithm with Convergence Analysis for Optimal Parameter Tuning,” *International Journal of Computational Intelligence and Applications*, vol. 16, no. 3, pp. 1750018, Sep. 2017. https://doi.org/10.1142/S1469026817500183

[12] X. Zhanga, K. F. Fongb, and S. Y. Yuena, “A Novel Artificial Bee Colony Algorithm for HVAC Optimization Problems,” *HVAC&R Research*, vol. 19, no. 6, pp. 715–731, 2013.

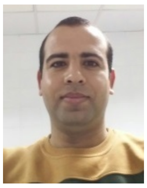
[13] M.-C. Niculescu, “Real-Time Control Of A Two Link Manipulator Using Multi-Layered Perceptron’s,” *Journal of Control Engineering and Applied Informatics*, vol. 4, pp. 49–54, 2003.

[14] J. G. Ziegler and N. B. Nichols, “Optimum Setting for Automatic Controllers,” *Trans. ASME*, vol. 64, pp. 759–768, 1942. https://doi.org/10.1115/1.2899060

[15] D. Karaboga and B. Basturk, “A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm,” *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Apr. 2007. https://doi.org/10.1007/s10898-007-9149-x

[16] Z. Bingül and O. Karahan, “A Fuzzy Logic Controller Tuned With PSO for 2 DOF Robot Trajectory Control,” *Expert Systems with Applications*, vol. 38, no. 1, pp. 1017–1031, Jan. 2011. https://doi.org/10.1016/j.eswa.2010.07.131

[17] N. A. Elkhateeb and R. I. Badr, “Dynamic Inertia Weight Artificial Bee Colony Versus GA and PSO for Optimal Tuning of PID Controller,” *International Journal of Modelling, Identification and Control*, vol. 22, no. 4, pp. 307, 2014. https://doi.org/10.1504/IJMIC.2014.066262



Nasr A. Elkhateeb obtained his B. sc., M. sc. degree with honours from the Electronics and Communications Department, Faculty of Engineering, Cairo University, in 2006 and 2011, respectively. He earned his Ph.D. in Electronics and Communications Engineering in 2017 from Cairo University. His research interests are advanced programming techniques, artificial intelligence, evolutionary programming techniques, and robotic control systems.
Address: Modern University for Technology and Information, Faculty of Engineering, ECE Dept., Cairo, Egypt.
E-mail: nasrantar@gmail.com

Ragia I. Badr obtained her B. sc. degree with distinction and honours in 1979 from the Electronics and Communications Department, Faculty of Engineering, Cairo University. She got her M. sc. degree in Systems and Control Engineering in 1982. She earned her Ph.D. in Systems and Control Engineering in 1987 from Cairo University. In 1987 she was appointed Assistant Professor in the Electronics and Communications Department, Faculty of Engineering, Cairo University. She is currently a Professor at the same department and her research areas include Robust Constrained Control, Neuro-Fuzzy Control, and Adaptive-Non-Linear Control.
Address: Cairo University, Faculty of Engineering, Giza, Egypt.
E-mail: ragiabadr@yahoo.com