

Research Article

Application of Improved BQGA in Robot Kinematics Inverse Solution

Xiaoqing Lv¹ and Ming Zhao ²

¹School of Electronics and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China

²School of Applied Technology, University of Science and Technology Liaoning, Anshan 114051, China

Correspondence should be addressed to Ming Zhao; as_zmyli@163.com

Received 26 October 2018; Accepted 9 January 2019; Published 3 February 2019

Academic Editor: Raffaele Di Gregorio

Copyright © 2019 Xiaoqing Lv and Ming Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the problem that Bloch Quantum Genetic Algorithm (BQGA) is easy to fall into local optimum, an improved BQGA is proposed. The algorithm can control the step size and the mutation probability in real time in the iterative process, avoiding over the optimal solution and guaranteeing search efficiency. In addition, the improved algorithm further completes the anti-degradation mechanism, which maintains the diversity of the population while preserving the dominant gene to the maximum extent, so that the algorithm is not easy to fall into the local extremum and finally approaches the global optimal solution. The application in the inverse solution of robot kinematics shows that the improved BQGA effectively avoids the premature problem and accelerates the convergence of understanding and the search result is close to the complete solution, which provides a new idea for solving complex nonlinear and multivariate functional equations.

1. Introduction

The kinematics problem is the basis for the intelligent development of today's robotic arms. Research in related fields has made tremendous contributions to the development of science and technology. In document [1] the solutions of both the forward and inverse kinematics problems for Adept Six 300 6R robot are presented. A matrix method was used in computations. Algorithm based on the derived formulas and logical conditions has been implemented in Adept Smart Controller Cx. Literatures [2] are based on kinematics for more in-depth path planning studies. This paper innovatively applies BQGA to the inverse kinematics program, which not only expands the application field of intelligent algorithms, but also provides a new perspective for solving kinematic problems.

In robotics [3], the positive solution problem is relatively simple and will get a unique solution, while the inverse solution problem [4] is relatively complex, and there may be multiple sets of solutions. The traditional methods of solving them are geometry and algebra methods [5, 6], numerical analysis method [7, 8], etc., but they are limited by

structure, space, expressions, etc., so it is not convenient and efficient to solve inverse kinematics problem. With the rapid development of computer programming technology, various intelligent algorithms, such as neural network algorithm [9], genetic algorithm [10, 11], particle swarm optimization [12], quantum optimization algorithms, and their various hybrid algorithms, have emerged, which highlight the ability to solve complex problems. For instance, in order to simplify the controller design of the manipulator, document [13] proposes to train the NN algorithm with entropy clustering method, which effectively solves the multi-value problem; applying an improved genetic algorithm to the inverse kinematics can overcome the structural limitations of the manipulator [14]; in the inverse solution of the EOD manipulator, the simulated annealing algorithm was used, and the expected effect was obtained [15]. With the development of quantum theory, people are increasingly interested in quantum optimization algorithms. Quantum Genetic Algorithm (QGA) is popular among scholars because of its convenience and efficiency. Therefore, many improved algorithms such as real-coded QGA [16], adaptive double-chain QGA [17], and Bloch Quantum Genetic Algorithm (BQGA) [18] have emerged.

Among them, BQGA is a probability optimization algorithm that uses the qubits to represent population elements [19]. Its three-chain coding method is more efficient than the traditional double-chain structure, which makes the GA algorithm take a new step. However, in practical applications, BQGA also has problems such as being easy to pass the optimal solution and population degradation, and sometimes the optimization result is not ideal. Therefore, this paper proposes an improvement plan for these problems in BQGA.

2. Improvement Scheme of BQGA

2.1. Adjustment of Adaptive Corner Value. In the process of updating BQGA's quantum chromosome, the value of the corner is more complicated. If the planning is unreasonable, it will seriously affect the optimization effect of the algorithm. The look-up table method in literature [18] provides an effective convergence strategy. However, the look-up table method does not take into account the differences in individual chromosomes and the trend of the evaluation function, making the algorithm easy to cross the optimal solution and less adaptive. Based on the traditional table lookup method, this paper further proposes an adaptive adjustment strategy for corner values. The expression is as follows:

$$|\Delta\varphi_{ij}| = \begin{cases} |\Delta\tilde{\varphi}_{ij}| & |\nabla f(X_{ib}^t)| \geq a \\ |\Delta\tilde{\varphi}_{ij}| & |\nabla f(X_{ib}^t)| < a \end{cases} \quad (1)$$

$$|\Delta\theta_{ij}| = \begin{cases} |\Delta\tilde{\theta}_{ij}| & |\nabla f(X_{ib}^t)| \geq a \\ |\Delta\tilde{\theta}_{ij}| & |\nabla f(X_{ib}^t)| < a \end{cases} \quad (2)$$

where t represents evolutionary algebra (same number of iterations), $|\Delta\varphi_{ij}|$ and $|\Delta\theta_{ij}|$ are the corner sizes of φ and θ of the j th qubit on the i rd chromosome, X_{ib}^t represents the optimal solution among the three chains of each chromosome in the t nd generation, $|\nabla f(X_{ib}^t)|$ represents the fitness gradient value of the optimal solution for each chromosome in the t nd generation, a is an evaluation value whose size is determined by the maximum and minimum gradient values in each generation, and $|\Delta\tilde{\varphi}_{ij}|$ and $|\Delta\tilde{\theta}_{ij}|$ are the corner values associated with the number of iterations, and the expression is

$$|\Delta\tilde{\varphi}_i| = \varphi_0 \times \frac{t_{\max} - t}{t_{\max}} \quad (3)$$

$$|\Delta\tilde{\theta}_i| = \theta_0 \times \frac{t_{\max} - t}{t_{\max}} \quad (4)$$

where φ_0 and θ_0 are initial iteration values, which range from 0.005π to 0.05π , and t_{\max} is the maximum number of iterations. $|\Delta\tilde{\varphi}_{ij}|$ and $|\Delta\tilde{\theta}_{ij}|$ are the angle values associated with the gradient. Referring to literature [18], the expressions constructed are as follows:

$$|\Delta\tilde{\varphi}_{ij}| = \varphi_0 \times \exp\left(-\frac{\nabla f(X_{ib}^t) - \nabla f^t \min}{\nabla f^t \max - \nabla f^t \min}\right) \quad (5)$$

$$|\Delta\tilde{\theta}_{ij}| = \theta_0 \times \exp\left(-\frac{\nabla f(X_{ib}^t) - \nabla f^t \min}{\nabla f^t \max - \nabla f^t \min}\right) \quad (6)$$

where ∇f^t_{\max} and ∇f^t_{\min} represent the maximum and minimum fitness gradient values of the generation, respectively. It should be pointed out that the discrete optimization problem [20] and the continuous optimization problem [21] are calculated differently when acquiring the relevant gradient. For the discrete optimization problem, since $f(X)$ does not have a gradient, we can replace the gradient with the difference of the fitness of two adjacent generations [22]; the expression is

$$\nabla f(X_{ib}^t) = |f(X_{ib}^t) - f(X_{ib}^{t-1})| \quad (7)$$

$$\nabla f^t \max = \max\{|f(X_{1b}^t) - f(X_{1b}^{t-1})|, \dots, |f(X_{mb}^t) - f(X_{mb}^{t-1})|\} \quad (8)$$

$$\nabla f^t \min = \min\{|f(X_{1b}^t) - f(X_{1b}^{t-1})|, \dots, |f(X_{mb}^t) - f(X_{mb}^{t-1})|\} \quad (9)$$

The gradient in the continuous optimization problem is

$$\nabla f(X_{ib}^t) = \left| \frac{\partial f(X_{ib}^t)}{\partial X_{ib}^t} \right| \quad (10)$$

$$\nabla f^t \max = \max\left\{\left| \frac{\partial f(X_{1b}^t)}{\partial X_{1b}^t} \right|, \dots, \left| \frac{\partial f(X_{mb}^t)}{\partial X_{mb}^t} \right|\right\} \quad (11)$$

$$\nabla f^t \min = \min\left\{\left| \frac{\partial f(X_{1b}^t)}{\partial X_{1b}^t} \right|, \dots, \left| \frac{\partial f(X_{mb}^t)}{\partial X_{mb}^t} \right|\right\} \quad (12)$$

In the above adaptive strategy, the gradient size (measured by a) determines the calculation formula of the corner. In the formula related to the gradient, the larger the gradient is, the smaller the step size becomes and, in the formula related to the number of iterations, the step size decreases monotonically with the increase of the number of iterations. Thus, the size of the corner varies with the gradient and is again limited by the number of iterations. The purpose is to control the step size in a reasonable range in real time.

2.2. Adaptive Mutation Probability. During the quantum chromosomal mutation of BQGA, the qubits rotate greatly along the Bloch sphere and are not constrained by other chromosomes, which helps to increase population diversity and avoid premature convergence. However, if the probability of mutation is constant throughout the evolution process, it is not conducive to the late convergence of the algorithm. In order to enable the algorithm to expand the global search ability in the early iteration and focus on local search in the late iteration to speed up the convergence, the adaptive mutation probability [23] is adjusted to

$$p_m = p_{m0} \times \frac{t_{\max} - t}{t_{\max}} \quad (13)$$

where $p_{m0} \in (0.01, 0.5)$.

2.3. Anti-Degradation Mechanism. In the process of population evolution, the anti-degradation mechanism can increase the chances of dominant individuals participating in inheritance and variation and reduce the possibility of population degradation, thus ensuring that the algorithm converges with maximum probability within a limited number of iterations. However, the principle of replacement of individual elimination will cause the population diversity to decrease rapidly with the increase of the number of iterations, so that it is easy to fall into local optimum, and it is difficult to jump out only by relying on mutation. In the improved BQGA, the replacement principle of the chain change is adopted. The description is that the chain is changed first, and the entire chromosome is replaced when the chain number is not the same. That is, it is the first to find the optimal solution (corresponding to the optimal chain) and the worst solution (corresponding to the worst chain) in one generation. If the chain numbers are the same, the worst chain is replaced by the optimal chain, and the other two chains remain unchanged; if the two chain numbers are different, replace the entire chromosome. The replacement expression is

$$\begin{aligned} \text{BAX} &\leftarrow \text{BEX} & \text{gbads} &= \text{gbests} \\ \text{BAC} &\leftarrow \text{BEC} & \text{gbads} &\neq \text{gbests} \end{aligned} \quad (14)$$

where BEX is the most suitable solution called the contemporary optimal solution, and its chromosome is called the contemporary optimal chromosome [24, 25] recorded as BEC. The same reason BAX is the worst solution and BAC is the worst chromosome, gbads and gbests represent the corresponding chain number. The anti-degradation mechanism of the improved BQGA ensures that the worst chromosomes of each generation have an optimal gene chain while maintaining population diversity.

3. Applying Improved BQGA for Inverse Kinematics

3.1. Establishment of Evaluation Function. Using the improved BQGA to solve the inverse kinematics problem of the robot, an evaluation function needs to be established. According to the kinematics principle of the n-degree-of-freedom manipulator [26], the kinematics positive solution equation established by the D-H parameter method [27, 28] is

$$\begin{aligned} \mathbf{A}^n &= \mathbf{A}_1^0(\omega_1) \cdot \mathbf{A}_2^1(\omega_2) \cdot \mathbf{A}_3^2(\omega_3) \cdot \mathbf{A}_4^3(\omega_4) \cdot \dots \\ &\cdot \mathbf{A}_n^{n-1}(\omega_n) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (15)$$

That is, given a set of angles from ω_1 to ω_n , matrix \mathbf{A}^n of position and posture of the end effector is uniquely determined. However, in practice it is the opposite; that is, the end position and posture are known to solve the value of the angle of each joint (inverse solution). In order to evaluate each

set of inverse solutions, we have established (16) [4, 29] which transforms the inverse solution problem into a multivariate function to find the extremum problem.

$$F = \|\mathbf{A}^n - \mathbf{T}\|_2 \quad (16)$$

where \mathbf{A}^n is the actual position and posture matrix of the robot arm established by the D-H parameter method, and \mathbf{T} is the set target position and posture matrix, $\|\mathbf{A}^n - \mathbf{T}\|_2$ is the two norm of \mathbf{A}^n and \mathbf{T} [30] which represents the distance between the two matrices.

Available from (16), when the actual position and posture of the robot arm are equal to or infinitely close to the target position and posture, the minimum value of the evaluation function F is zero or approximately zero.

3.2. Coding Method. According to the basis of quantum computing and the three-strand gene coding scheme of quantum chromosomes [18, 31], the initial population expression is as follows:

$$q_i^t = \begin{bmatrix} \cos \varphi_{i1}^t \sin \theta_{i1}^t & \cos \varphi_{i2}^t \sin \theta_{i2}^t & \dots & \cos \varphi_{in}^t \sin \theta_{in}^t \\ \sin \varphi_{i1}^t \sin \theta_{i1}^t & \sin \varphi_{i2}^t \sin \theta_{i2}^t & \dots & \sin \varphi_{in}^t \sin \theta_{in}^t \\ \cos \theta_{i1}^t & \cos \theta_{i2}^t & \dots & \cos \theta_{in}^t \end{bmatrix} \quad (17)$$

where $\varphi_{ij} \in (\pi/2, 3\pi/2)$, $\theta_{ij} \in (0, \pi)$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$; m is the population size, which is the number of chromosomes; n is the number of qubits (representing n to-be-turned corners); t is an evolutionary algebra.

3.3. Transformation of Unit Space Solution to Joint Space Solution. According to the theory of Bloch spherical coordinates, a population can represent the m-group solutions of the search space and map them into the actual solution space $\Omega = [-2\pi, 2\pi]^n$ to obtain m sets of feasible solutions (X_{ix} , X_{iy} , and X_{iz}). The mapping formula [18] is as follows:

$$X_{ix}^j = \frac{1}{2} [V_{\max} (1 + p_{ix}^j) + V_{\min} (1 - p_{ix}^j)] \quad (18)$$

$$X_{iy}^j = \frac{1}{2} [V_{\max} (1 + p_{iy}^j) + V_{\min} (1 - p_{iy}^j)] \quad (19)$$

$$X_{iz}^j = \frac{1}{2} [V_{\max} (1 + p_{iz}^j) + V_{\min} (1 - p_{iz}^j)] \quad (20)$$

where $j = 1, 2, \dots, n$; V_{\max} and V_{\min} correspond to the maximum and minimum values in the range of values for each angle (check Table 1 to get its value); p_{ix}^j , p_{iy}^j , and p_{iz}^j represent the X solution, the Y solution, and the Z solution of the j th gene position of the i th chromosome, respectively. According to the decoding process, each of the chains of each chromosome decodes n corresponding robot arm corner values and their values directly fall within the respective range of values.

3.4. Update and Mutation of Quantum Chromosomes. The quantum chromosome is updated by the quantum revolving

TABLE 1: Connecting rod parameters of PUMA560.

i	$\alpha_{i-1}(\text{rad})$	$a_{i-1}(\text{m})$	$d_i(\text{m})$	$\omega_i(\text{rad})$	Joint angle range(rad)
1	0	0	0	ω_1	$-8\pi/9 \sim 8\pi/9$
2	$-\pi$	0	d_2	ω_2	$-5\pi/4 \sim \pi/4$
3	0	a_2	0	ω_3	$-\pi/4 \sim 5\pi/4$
4	$-\pi$	a_3	d_4	ω_4	$-11\pi/18 \sim 17\pi/18$
5	π	0	0	ω_5	$-5\pi/9 \sim 5\pi/9$
6	$-\pi$	0	0	ω_6	$-133\pi/90 \sim 133\pi/90$

where $a_2 = 0.4318$, $a_3 = 0.02032$, $d_2 = 0.14909$, $d_4 = 0.43307$.

TABLE 2: Optimization results of BQGA and improved BQGA.

Optimization results	$\omega_1(\text{rad})$	$\omega_2(\text{rad})$	$\omega_3(\text{rad})$	$\omega_4(\text{rad})$	$\omega_5(\text{rad})$	$\omega_6(\text{rad})$
BQGA	-1.2877	-3.5668	2.8165	1.9572	-1.2525	3.0360
Improve BQGA	-1.1290	-3.4995	2.8803	2.0680	-1.2586	3.1212

TABLE 3: Comparison of convergence between basic BQGA and improved BQGA.

Number of iterations	50	100	200	400	800	1000
The F value of BQGA	0.0716	0.0643	0.0479	0.0373	0.0373	0.0373
The F value of improve BQGA	0.0273	0.0210	0.0148	0.0085	0.0081	0.0071

gate [24, 32], which causes the current chromosome to evolve toward the direction of the contemporary optimal chromosome, producing the next generation. The symbol of the two corners of the quantum revolving door is selected in [18], and the size is determined by using the adaptive corner value adjustment scheme of Section 2.1. It should be pointed out that since each corner of the robot arm has a specific range of values, the F formed by it cannot be continuously taken. Therefore, the inverse solution is a discrete optimization problem, and it is necessary to replace the gradient with two generations of difference.

The mutation of quantum chromosomes is realized by the non-gate [24, 33], and its role can be seen as an unconstrained large rotation. The adaptive mutation probability scheme in Section 2.2 is used to control the usage of non-gates.

3.5. Steps to Find Inverse Kinematics. Based on the above-mentioned theory of improving BQGA, the flow chart for finding the inverse kinematics is shown in Figure 1.

4. Application Example of Improved BQGA

In this paper, the general PUMA560 is taken as an example, and the inverse solution is obtained by using BQGA and improved BQGA, respectively. It is known that the connecting rod parameters [34, 35] of the PUMA 560 are as shown in Table 1. In the operable space, it is assumed that the target position and posture matrix \mathbf{T} are randomly generated as

$$\mathbf{T} = \begin{bmatrix} 0.5322 & -0.1137 & -0.8390 & 0.0833 \\ -0.4697 & -0.8641 & -0.1809 & 0.1890 \\ -0.7044 & 0.4903 & -0.5133 & -0.4925 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (21)$$

According to the flowchart of the inverse kinematics solution in Figure 1, the initial parameters are $m = 100$; $n = 6$; $p_{m0} = 0.1$; $\varphi_0 = 0.05\pi$; $\theta_0 = 0.05\pi$; $t_{\max} = 1000$. The program is written in Matlab. Under the same initial conditions, the optimization results of BQGA and improved BQGA are shown in Table 2. The convergence is shown in Table 3 and Figure 2.

The values of the two sets of optimal solution in Table 2 are substituted into the kinematics positive equation (15), and the actual position and posture matrix of the BQGA and the improved BQGA are \mathbf{A}_1 and \mathbf{A}_2 , respectively.

$$\mathbf{A}_1 = \begin{bmatrix} 0.4988 & -0.1207 & -0.8583 & 0.1199 \\ -0.5406 & -0.8174 & -0.1992 & 0.1215 \\ -0.6775 & 0.5633 & -0.4730 & -0.4811 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (22)$$

$$\mathbf{A}_2 = \begin{bmatrix} 0.5336 & -0.1143 & -0.8379 & 0.0764 \\ -0.4743 & -0.8608 & -0.1846 & 0.1872 \\ -0.7002 & 0.4959 & -0.5136 & -0.4921 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \quad (23)$$

It can be seen that the deviation of the corresponding element values in \mathbf{A}_2 and \mathbf{T} is smaller. That is, the optimal solution of the improved algorithm is closer to the actual solution of the target equation (this can also be seen from the F value corresponding to the same number of iterations in Table 3), indicating that the actual position and posture of the end effector of the mechanical arm with improved BQGA can be closer to the set target pose.

It can be seen from Figure 2 and Table 3 that after the BQGA is iterated 250 times, the F value does not change,

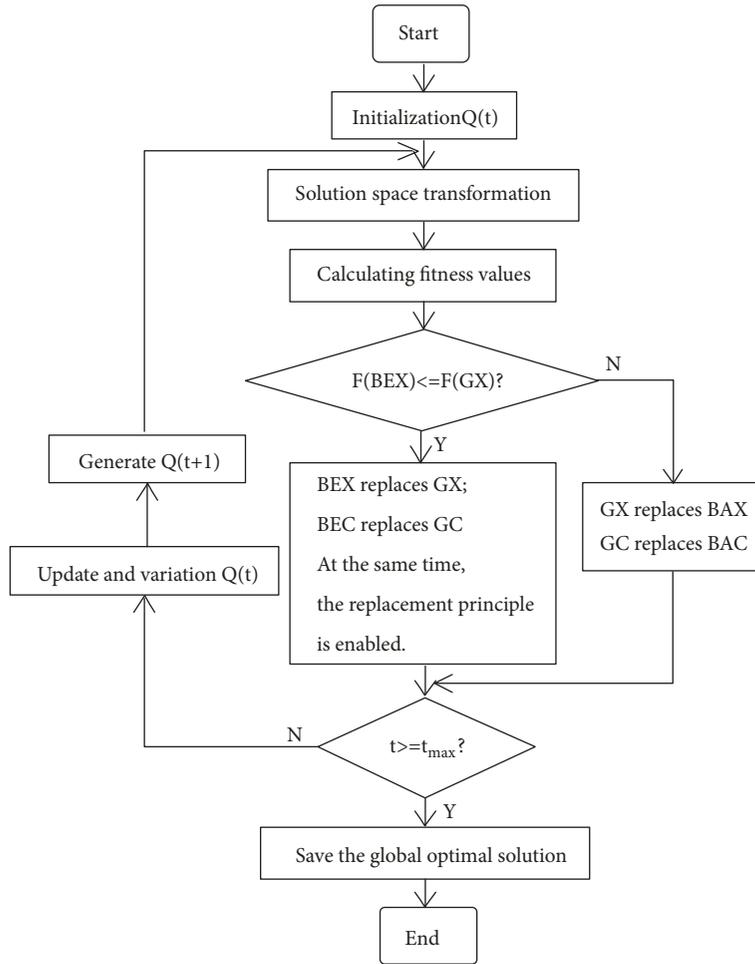


FIGURE 1: Flow chart for finding inverse kinematics. where, as for calculating the fitness value with the evaluation function F , the smaller F is, the greater fitness is; GX and GC represent the global optimal solution and the global optimal chromosome, respectively.

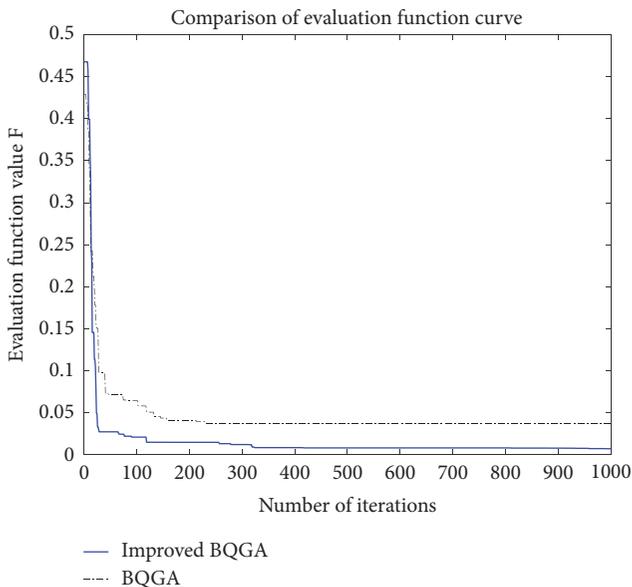


FIGURE 2: Comparison of evaluation function curve.

indicating that it has fallen into the local optimal solution. Not only does the improved BQGA find faster speed, but also the curve gradually approaches the zero axis with the increase of the number of iterations, which highlights the superiority of the improved algorithm not easily falling into the local extremum.

5. Conclusions

(i) The improved BQGA solves the problems of the BQGA better and has the advantages of high search efficiency, difficulty in overcoming the optimal solution, and difficulty in falling into local extremum.

(ii) The improved BQGA not only solves the problem of inverse kinematics of robots, but also effectively improves the speed and accuracy of robotic control. It provides a method for solving similar complex multivariate functions.

(iii) Although this paper solves the typical discrete optimization problem, it is instructive and versatile for some continuous optimization and other fields of application.

Data Availability

The data used to support the findings of this study are included within the article. The source program used to support the findings of this study is included within the supplementary information file (available here).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Supplementary Materials

The source program used to support the findings of this study is included within the supplementary information file. (*Supplementary Materials*)

References

- [1] A. Legowski, "The global inverse kinematics solution in the adept six 300 manipulator with singularities robustness," in *Proceedings of the 20th International Conference on Control Systems and Computer Science, CSCS 2015*, pp. 90–97, Romania, 2015.
- [2] A. Legowski, M. Niezabitowski, and T. Grzeszczak, "Orientational path planning for 6R robot," in *Proceedings of the 17th IEEE International Carpathian Control Conference, ICC 2016*, pp. 434–439, Slovakia, June 2016.
- [3] Z. X. Cai, *Robotics foundation*, Mechanical Industry Press, Beijing, China, 2015.
- [4] M. Zhao and Y. Dai, "Application of fuzzy ant colony algorithm to robotics arm inverse kinematics problem," *ICIC Express Letters*, vol. 10, no. 1, pp. 43–49, 2016.
- [5] X. Li, Y. Guo, J. Zhang, and S. Guo, "Inverse kinematics solution and verification of modular 6-DOF manipulator," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 44, no. 4, pp. 246–251, 2013.
- [6] H.-C. Jiang, S.-R. Liu, and B.-T. Zhang, "Inverse kinematics analysis for 6 degree-of-freedom modular manipulator," *Journal of Zhejiang University(Engineering Science)*, vol. 44, no. 7, pp. 1348–1354, 2010.
- [7] Y. J. Liu, "Inverse kinematics and trajectory planning of 6R serial manipulators," *Journal of Mechanical Engineering*, vol. 48, no. 03, pp. 9–15, 2012.
- [8] H. Ananthanarayanan and R. Ordóñez, "Real-time Inverse Kinematics of $(2n + 1)$ DOF hyper-redundant manipulator arm via a combined numerical and analytical approach," *Mechanism and Machine Theory*, vol. 91, pp. 209–226, 2015.
- [9] W. Wang, S. Zhang, and S. Yu, "Image Resteration by BP Neural Based on PSO," *Journal of Northwestern Polytechnical University*, vol. 36, no. 4, pp. 709–714, 2018.
- [10] Y. LIN, "Solution of Inverse Kinematics for General Robot Manipulators Based on Multiple Population Genetic Algorithm," *Journal of Mechanical Engineering*, vol. 53, no. 3, p. 1, 2017.
- [11] S. Momani, Z. S. Abo-Hammour, and O. M. K. Alsmadi, "Solution of inverse kinematics problem using genetic algorithms," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, pp. 225–233, 2016.
- [12] C. Ji, C. F. Shan, Y. Sha, and R. K. Zhou, "Blind source separation based on grouping simplified particle swarm optimization algorithm," *Journal of Northeastern University (Natural Science)*, vol. 39, no. 6, pp. 787–791, 2018.
- [13] C. Zhang, J. F. Yang, and F. Zhang, "Manipulator kinematics inverse solution study based on EC-RBF learning algorithm," *Manufacturing Automation*, vol. 35, no. 14, pp. 64–65+72, 2013.
- [14] J. L. Wang, G. L. Zhang, B. Jing, and J. Xv, "Research on way of inverse kinematic solution of six DOF manipulator," *Computer Engineering and Applications*, vol. 49, no. 22, pp. 266–270, 2013.
- [15] Y. F. Zhang and Z. S. Ma, "A simulated annealing neural network solution to inverse kinematics of EOD manipulator," *Computer Measurement & Control*, vol. 23, no. 4, pp. 1269–1272, 2015.
- [16] J. Liu, H. Wang, Y. Sun, C. Fu, and J. Guo, "Real-Coded Quantum-Inspired Genetic Algorithm-Based BP Neural Network Algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 571295, 10 pages, 2015.
- [17] H. Kong, N. Li, and Y. Shen, "Adaptive double chain quantum genetic algorithm for constrained optimization problems," *Chinese Journal of Aeronautics*, vol. 28, no. 1, pp. 214–228, 2015.
- [18] S. Y. Li and P. C. Li, *Quantum computation and quantum optimization algorithms*, Harbin Institute of Technology Press, Harbin, China, 2009.
- [19] R. Nowotniak and J. Kucharski, "Higher-order quantum-inspired genetic algorithms," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014*, pp. 465–470, Poland, 2014.
- [20] G. S. Yin, X. H. Cui, Y. X. Dong, and X. Yang, "An improved binary particle swarm optimization for discrete optimization problems," *Journal of Harbin Engineering University. Harbin Gongcheng Daxue Xuebao*, vol. 36, no. 2, pp. 191–195, 2015.
- [21] Q. Zhang and P.-C. Li, "Adaptive grouping difference firefly algorithm for continuous space optimization problems," *Control and Decision*, vol. 32, no. 7, pp. 1217–1222, 2017.
- [22] R. Zheng, *Convergence analysis and improvement method of Double-Chain quantum genetic algorithm*, Nanchang Aviation University, 2012.
- [23] J. C. Li, K. Han, and T. Z. Bao, "An improved bloch spherical quantum genetic algorithm and its application," *Journal of Railway Science and Engineering*, vol. 13, no. 11, pp. 2262–2269, 2016.
- [24] P. C. Li, *Quantum computation with applications to intelligent optimization and control*, Harbin Institute of Technology, 2009.
- [25] Z. J. Yi, K. Hou, and R. H. He, "Adaptive quantum genetic algorithm based on Bloch sphere," *Computer Engineering and Applications*, vol. 48, no. 35, pp. 57–61, 2012.
- [26] X. Han, M. Yin, X. Liu, and G. Yin, "Solution of inverse kinematics and movement trajectory simulation for 6R robot," *Journal of Sichuan University(Engineering Science Edition)*, vol. 47, no. 6, pp. 185–190, 2015.
- [27] B.-Q. ian, J.-C. u, and A.-Q. hang, "Dynamic modeling of wave driven unmanned surface vehicle in longitudinal profile based on D-H approach," *Journal of Central South University*, vol. 22, no. 12, pp. 4578–4584, 2015.
- [28] H. M. Herath, R. A. Gopura, and T. D. Lalitharatne, "An Underactuated Linkage Finger Mechanism for Hand Prostheses," *Modern Mechanical Engineering*, vol. 08, no. 02, pp. 121–139, 2018.
- [29] Y. Dai and M. Zhao, "Manipulator path-planning avoiding obstacle based on screw theory and ant colony algorithm," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 1, pp. 922–927, 2016.

- [30] J. G Zhang and X. F Zhang, *Matrix Analysis and Calculation*, Wuhan University Press, Wuhan, China, 2013.
- [31] F. F. Qian and F. Xu, "Quantum evolutionary algorithm based on Bloch sphere," *Software Guide*, vol. 12, no. 2, pp. 58–60, 2013.
- [32] J. Z. Ji, L. Cheng, X. W. Zhao, and C. N. Liu, "Quantum ant colony algorithm for the multi-task coalition problem," *Journal of Beijing University of Technology. Beijing Gongye Daxue Xuebao*, vol. 39, no. 3, pp. 412–419, 2013.
- [33] Y. Luo and J. Duo, "Multi-objective reactive power optimization based on quantum immune colonial algorithm," *Electric Power Automation Equipment*, vol. 33, no. 9, pp. 31–35, 2013.
- [34] Y. Li, W. Z. Zhang, P. Lin, and W. J. Li, "The application of RBF networks in inverse kinematic of PUMA560 robot," *Machinery Design & Manufacture*, vol. 7, pp. 94–96, 2009.
- [35] M. Sha and Z. L. Deng, "Study of space of PUMA560 robot based on matlab," *Machinery Manufacturing Automation*, vol. 45, no. 2, pp. 156–159, 2016.

