U.P.B. Sci. Bull., Series D, Vol. 83, Iss. 4, 2021 ISSN 1454-2358

APPLICATION OF IMPROVED MULTI-POPULATION ADAPTIVE GENETIC ALGORITHM FOR SOLVING THE INVERSE KINEMATICS OF REDUNDANT MANIPULATOR

Xinglei ZHANG[1, 3*], Binghui FAN[2*], Chuanjiang WANG[2], Xiaolin CHENG[4]

Due to residual redundancy, the redundant manipulators have infinite joint positions to satisfy the desired pose of the end effector, but a unique inverse solution is very difficult to obtain. In this paper, an improved multi-population adaptive genetic algorithm is proposed to solve this problem. Firstly, the inverse solution problem is transformed into an optimization problem. Then, aiming at the shortcomings of the traditional multi-population genetic algorithms in the process of solving inverse kinematics, some improvement measures are presented to increase the accuracy and speed of inverse kinematics solution. Finally, our algorithm is simulated in a redundant manipulator, and compared with the traditional multi-population genetic algorithm. The simulations indicate that the average pose error obtained by our algorithm is 0.15mm and 0.00023rad which are better than 3.89mm and 0.051rad obtained by the traditional multi-population genetic algorithm. The average number of iteration steps by our algorithm is 53, which is faster than 114 obtained by the traditional multi-population genetic algorithm.

Key words: genetic algorithm, inverse kinematics, objective function, redundant manipulator.

1. Introduction

Compared with 6-degree-of-freedom (DOF) manipulators, redundant manipulators have some advantages under the premise of satisfying the end effector (EE) tasks, such as singularity avoidance and obstacle avoidance etc, but the inverse kinematics (IK) solution is more complex. How to get the appropriate solution according to the actual task is a critical issue. The solving methods include closed methods (applicable on particular cases) and numerical methods [1]. For redundant manipulators that do not meet Pieper criterion [2] can only

[1]College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao, 266590, China.
[2]College of Electrical and Automation Engineering, Shandong University of Science and Technology, Qingdao, 266590, China.
[3]College of Mechanical and Electronic Engineering, Zaozhuang University, Zaozhuang, 277160, China.
[4]College of Mechanical Engineering, Shandong University, Jinan, 250100, China.
*Email: zhangxingfulove@163.com, fbhzxl@126.com.

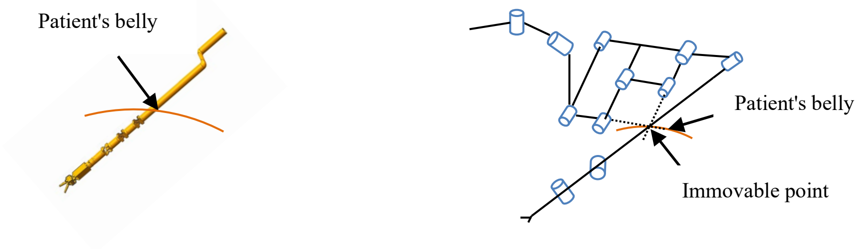
obtain IK solution by numerical methods [3] or intelligent algorithms. The current commonly used numerical method is iterative algorithm, which is to calculate the Jacobian pseudo-inverse matrix of the redundant manipulators, then iteratively calculate the exact solution through the given initial value. However, this algorithm has a huge amount of calculation, and when the redundant manipulators move to the Jacobian singular configuration, the inverse Jacobian matrix becomes numerically unrealizable and will cause the joint velocities to tend to infinity, which is not allowed for robotic manipulators.

In recent years, intelligent algorithms [4] have been widely used to solve the IK problem. The main idea is to transform kinematics equation of robots into optimization problem to obtain the optimal solution of IK, including genetic algorithm, immune algorithm, particle swarm optimization, etc. Gao Ruihua [5] proposed a robotics IK solution algorithm based on improved backpropagation neural network (BPNN), but a lot of network training is needed and only used in the training environment. Jiang Guanwu [6] designed a particle swarm optimization combined BPNN algorithm to obtain the IK solution. Although it overcame some disadvantages of BPNN, the accuracy was not stable enough. Tang Yuanhui [7] proposed an artificial fish swarm algorithm to find the solution of IK. Paulo Costa [8] provided the stretched simulated annealing method and the NSGA Ⅱ method to determine a solution for the IK problem of a 12-DOF robotic arm. Cai Zefan [9] proposed a particle swarm optimization algorithm to obtain the IK solution of multi-joint robot. However, the number of iterations was large and the convergence speed was slow. Ren Ziwu [10] designed a hybrid biogeography-based optimization method for the IK problem of a 8-DOF redundant humanoid manipulator. Although the accuracy of solution is improved, the number of iteration steps is over 1000, which is not conducive to real-time control. Gerardo Pachicano [11] proposed a smart optimization for biomechanical linkages. It is computed using EvoNorm by minimizing a fitness function based on the joints position. But the processing time of this algorithm is 1h, which is not also conducive to real-time control. Z¨uhnja Riekert [12] researched a vector evaluated particle swarm optimization to solve the IK problem of redundant manipulators, but the accuracy of solution depends on the selection of weights. Wang Jiesheng [13] designed a grey wolf optimizer method based on differential evolution and elimination mechanism, but this algorithm is not mature enough and easy to fall into local optimum. Compared to the above intelligent algorithms, genetic algorithm is an optimization method based on natural selection mechanism in the process of biological evolution [14-16]. It has strong robustness and global optimization ability. The genetic algorithm has been widely used in inverse kinematics of robots because of its strong search ability, global convergence and avoidance of tedious formula derivation. Lin Ming [17] used an improved single population genetic algorithm to solve the IK problem, which reduces the evolutionary algebra and improves the convergence speed and accuracy. However, as the degree of freedom increases, the algorithm is easy to converge locally, so the convergence speed and accuracy are difficult to guarantee. For this reason, Lin Yang [18] proposed an IK solution method of multi-population genetic algorithm. Although the convergence speed has been improved, when constructing fitness function, the selection of position weight coefficients lacks theoretical basis, and the solution accuracy is unstable. Lv Xiaoqing and Ming Zhao [19] designed an improved bloch quantum genetic algorithm which was applied the algorithm to the robot IK solution, and the speed and accuracy are considerable.

To avoid shortcomings of iterative method and the traditional multi-population genetic algorithm (TMPGA), an improved multi-population adaptive genetic algorithm is proposed. This paper is organized as follows. Kinematics analysis is briefly reviewed in section 2. An IK algorithm is proposed in section 3. In section 4, the simulation analysis of the proposed algorithm is carried out in a redundant manipulator, and compared with the TMPGA to verify the effectiveness of our algorithm. Finally, the conclusions are presented in section 5.

2. Kinematic analysis

In this paper, a 7-DOF redundant manipulator which is designed by our research for abdominal surgery is used to kinematics analysis. This manipulator consists of a prismatic joint and six revolute joints as shown in Fig. 1 (a). The prismatic joint can move up and down along the incision, and all joints are placed in the abdominal cavity. Compared with the Da Vinci surgical manipulator (schematic diagram shown in Fig. 1 (b)), the 7-DOF redundant manipulator avoids the disadvantages of the immovable point and improves the working space and operating flexibility. The coordinate systems are established as shown in Fig. 2, and the modified D-H parameters are shown in Table 1.



(a) (b)

Fig. 1. (a) A 7-DOF redundant manipulator, (b) Simple schematic diagram of Da Vinci surgical manipulator

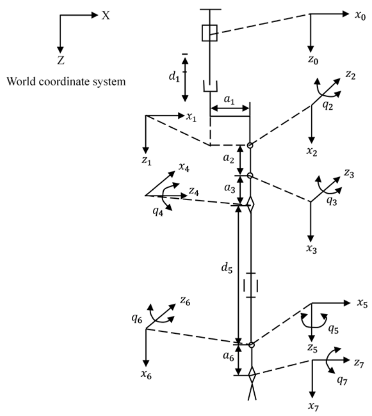


Fig. 2. A 7-DOF redundant manipulator coordinate system

Table 1

The modified D-H parameters of the 7-DOF redundant manipulator

Link	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)	$[\theta_{imin}, \theta_{imax}]$
1	0	0	d_1	0	[-100, 100] (mm)
2	$\pi/2$	20	0	$\theta_2(\pi/2)$	$[-\pi/2, \pi/2]$ (rad)
3	0	50	0	$\theta_3(0)$	$[-\pi/2, \pi/2]$ (rad)
4	$\pi/2$	50	0	$\theta_4(\pi/2)$	$[-\pi/2, \pi/2]$ (rad)
5	$\pi/2$	0	100	$\theta_5(\pi/2)$	$[-\pi, \pi]$ (rad)
6	$\pi/2$	0	0	$\theta_6(\pi/2)$	$[-\pi/2, \pi/2]$ (rad)
7	$\pi/2$	20	0	$\theta_7(0)$	$[-\pi/2, \pi/2]$ (rad)

The transformation matrix between adjacent coordinate systems is:

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & -d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

The forward kinematics ${}^0T_7$ expresses the pose (position and orientation) of the EE with respect to the base coordinate system:

$$ {}^0T_7 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4\,{}^4T_5\,{}^5T_6\,{}^6T_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & P \\ 0 & 1 \end{bmatrix} \quad (2) $$

Where $R_{3\times3}$ is the orientation matrix, P is the position vector.

The desired pose of the EE is given:

$$P_{qw}=\begin{bmatrix} R_{3\times3,qw} & P_{qw} \\ 0 & 1 \end{bmatrix} \tag{3}$$

Bringing the current joint positions into Eq. (2) to obtain the current pose of the EE：

$$P_{dq}=\begin{bmatrix} R_{3\times3,dq} & P_{dq} \\ 0 & 1 \end{bmatrix} \tag{4}$$

To make the EE of the manipulator reach the desired pose, it should satisfy the condition $\left|P_{dq}-P_{qw}\right|=0$. According to the theory of optimal design, the IK problem is decomposed into a nonlinear transcendental vector function and a constraint function. The IK solution is transformed into the following optimization control problem.

$$Min\left|P_{dq}-P_{qw}\right| = \begin{vmatrix} \Delta R_{3\times3} & \Delta P \\ 0 & 1 \end{vmatrix}$$

$$\text{Subject to } \theta_i \in [\theta_{\text{imin}}, \theta_{\text{imax}}] \tag{5}$$

3. Inverse kinematics solution method

3.1 Problems with Genetic Algorithms

Single-population genetic algorithm [20] is based on natural selection mechanism in the process of biological evolution. When the number of variables increases, there are some shortcomings due to the difficulty of coding and the sudden increase of computation, such as premature convergence, poor local search ability and serious assimilation of late population etc. Therefore, multi-population genetic algorithm has been designed to solve complex optimization problems (multi-peak and high-dimensional control etc) [21-22].

However, in the multi-population genetic algorithm, the selections of position weight coefficient of fitness function, encoding method, crossover method, crossover operator and mutation operator are all unreasonable, which will reduce the convergence speed and accuracy of the algorithm. Therefore, an improved multi-population adaptive genetic algorithm based on the TMPGA is proposed to solve these unreasonable aspects.

3.2 Improvement measures based on the TMPGA

1) Introduce the concept of matrix similarity.

According to the objective function, the result of optimization is to make $\Delta R_{3\times3}$ and ΔP to be zero. To achieve this goal, the matrix similarity is firstly defined as follows:

$$d = \begin{vmatrix} \Delta R_{3\times3} & \Delta P \\ 0 & 1 \end{vmatrix} = \begin{bmatrix} |n_{x,dq}-n_{x,qw}| & |o_{x,dq}-o_{x,qw}| & |a_{x,dq}-a_{x,qw}| & |p_{x,dq}-p_{x,qw}| \\ |n_{y,dq}-n_{y,qw}| & |o_{y,dq}-o_{y,qw}| & |a_{y,dq}-a_{y,qw}| & |p_{y,dq}-p_{y,qw}| \\ |n_{z,dq}-n_{z,qw}| & |o_{z,dq}-o_{z,qw}| & |a_{z,dq}-a_{z,qw}| & |p_{z,dq}-p_{z,qw}| \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Where n,o,a is the component of the orientation matrix in the x,y,z direction of the base coordinate system, respectively. When all elements of matrix d are infinitely close to zero, it indicates that P_{qw} and P_{dq} is highly similar. Because the first three rows of d are all non-negative values, so Eq. (6) can be written as:

$$\begin{aligned} d^* = & |n_{x,dq}-n_{x,qw}| + |n_{y,dq}-n_{y,qw}| + |n_{z,dq}-n_{z,qw}| + |o_{x,dq}-o_{x,qw}| + |o_{y,dq}-o_{y,qw}| + |o_{z,dq}-o_{z,qw}| \\ & + |a_{x,dq}-a_{x,qw}| + |a_{y,dq}-a_{y,qw}| + |a_{z,dq}-a_{z,qw}| + |p_{x,dq}-p_{x,qw}| + |p_{y,dq}-p_{y,qw}| + |p_{z,dq}-p_{z,qw}| \end{aligned} \tag{7}$$

2) Transform objective function.

Based on the principle of pose separation, the objective function is decomposed into two non-linear functions: position similarity function $f_1(x)$ and orientation similarity function $f_2(x)$. The ranges of each joint position are selected as the constraint function.

$$f_1(x) = |\Delta P| = |\Delta P_x| + |\Delta P_y| + |\Delta P_z| = |p_{x,dq}-p_{x,qw}| + |p_{y,dq}-p_{y,qw}| + |p_{z,dq}-p_{z,qw}| \tag{8}$$

$$\begin{aligned} f_2(x) = & |\Delta R_{3\times3}| = |\Delta n_x| + |\Delta o_y| + |\Delta a_z| = |n_{x,dq}-n_{x,qw}| + |n_{y,dq}-n_{y,qw}| + |n_{z,dq}-n_{z,qw}| \\ & + |o_{x,dq}-o_{x,qw}| + |o_{y,dq}-o_{y,qw}| + |o_{z,dq}-o_{z,qw}| + |a_{x,dq}-a_{x,qw}| + |a_{y,dq}-a_{y,qw}| + |a_{z,dq}-a_{z,qw}| \end{aligned} \tag{9}$$

$|\Delta P|$ and $|\Delta R_{3\times3}|$ can be regarded as the position error and orientation error of the EE, respectively

3) Construct fitness function.

In the process of optimization, genetic algorithm uses fitness function to evaluate individual's fitness. The fitness function should have the characteristics of single value, continuity, non-negative and maximization. Since the objective function is decomposed into two non-linear functions, according to the parameters of the manipulator, the ranges of $|\Delta P_x|$, $|\Delta P_y|$, $|\Delta P_z|$ are (0, 480), (0, 280), (0, 240) in the process of evolution, respectively. However, the ranges of

$\left|\Delta n_x\right|$, $\left|\Delta o_y\right|$, $\left|\Delta a_z\right|$ are all (0, 1). The large difference in magnitude causes genetic algorithm to ignore the orientation changes during the optimization process, which can lead to false similarity and local convergence. Therefore, weight coefficients should be introduced to make $\left|\Delta P_x\right|$, $\left|\Delta P_y\right|$, $\left|\Delta P_z\right|$ between 0 and 1, respectively. This paper uses min-max scaling method to obtain the weight coefficients of $\left|\Delta P_x\right|$, $\left|\Delta P_y\right|$, $\left|\Delta P_z\right|$. The min-max scaling method is as follows:

$$m^* = \frac{m - \min}{\max - \min} \tag{10}$$

Where m^* is the normalized data; m is the original data; max represents the maximum value of sample data; min represents the minimum value of sample data. Therefore, fitness function is constructed as follows:

$$Fitness = \frac{1}{1 + \mu_1 \cdot f_1(x)_x + \mu_2 \cdot f_1(x)_y + \mu_3 \cdot f_1(x)_z + \beta \cdot f_2(x)} \tag{11}$$

Where $f_1(x)_x = \left|\Delta P_x\right|, f_1(x)_y = \left|\Delta P_y\right|, f_1(x)_z = \left|\Delta P_z\right|$,

$\mu_1 = \frac{1}{480}$, $\mu_2 = \frac{1}{280}$, $\mu_3 = \frac{1}{240}$, $\beta = 1$.

4) Choose encoding method.

When binary coding is used in the traditional genetic algorithms, the "Heming Cliff" problem may occur. "Hamming Cliff" refers to the phenomenon that the discrete values of individuals in a population are very similar in phenotype, but are far apart in genotype. Because the object of genetic algorithm search is the encoding of individual variables, not the individual variables themselves. If we consider those solutions that are similar in phenotype to the optimal solution but very different in genotype as individuals with the optimal solution, then the calculated solution is not the true optimal solution [23]. In this paper, real-number coding that does not require specific coding and decoding is used to solve this problem and improve the efficiency of the algorithm.

5) Choosing cross operator.

Single-point crossover which cannot extract part of the genetic information before and after the individual is used in the Single-population genetic algorithm, so it easily results in a decline in search ability. The TMPGA uses multi-point crossover, which can make the search stable. However, it needs too many iteration steps to cause slow convergence rate. Well-distributed crossover can reduce the deviation between coding length and given parameter coding, and make the search more efficient. Therefore, well-distributed crossover combined multi-point crossover is used to improve the search stability and efficiency in this paper.

6) Determining crossover probability P_c and mutation probability P_m.

P_c and P_m of single population genetic algorithm are fixed values, and P_c and P_m of the TMPGA are random generated in [0.4, 0.9] and [0.005, 0.1]. The selective methods of P_c and P_m are very unreasonable, which affects the accuracy of the algorithm. The principle of adjustment of P_c and P_m is proposed as follows. In the early stage of evolution, the fitness of individuals is small, P_c should be greater and P_m should be smaller. As the iteration proceeds, the individual gradually becomes better, P_c should become smaller and P_m should become larger. This improvement measure is helpful to protect the fine individuals in the population and find the global optimum [24-26].

The adaptive genetic algorithm presented by Srinivas [27] can automatically change P_c and P_m with the fitness value, but when the individual fitness value is close to the maximum fitness value, P_c and P_m are close to zero. At this moment, the genetic algorithm is invalid. Ren Ziwu [28] proposed an improved adaptive genetic algorithm, which makes the individual fitness close to the maximum fitness value, P_c and P_m are far from zero, so that the algorithm can jump out of the local optimal solution. When the fitness value is greater than the average fitness value, P_c and P_m have the ability of automatic adjustment. However, when the fitness value is less than the average fitness value, P_c and P_m are fixed. Therefore, P_c and P_m are not automatically adjusted in the whole population, which is obviously inconsistent with the original intention of the adaptive genetic algorithm.

To solve the above problems, novel P_c and P_m which are automatically adjusted in [0.4, 0.9] and [0.005, 0.1] are proposed in the whole population.

$$P_c = \begin{cases} \dfrac{P_{c1} + P_{c2}}{2} + \dfrac{\left(P_{c1} - P_{c2}\right)\left(f_{avg} - f'\right)^2}{2\times\left(f_{avg} - f_{min}\right)^2} & f' < f_{avg} \\ \dfrac{P_{c1} + P_{c2}}{2} - \dfrac{\left(P_{c1} - P_{c2}\right)\left(f' - f_{avg}\right)^2}{2\times\left(f_{max} - f_{avg}\right)^2} & f' \geqslant f_{avg} \end{cases} \tag{12}$$

$$P_m = \begin{cases} \dfrac{P_{m1} + P_{m2}}{2} - \dfrac{\left(P_{m1} - P_{m2}\right)\left(f_{avg} - f\right)^2}{2\times\left(f_{avg} - f_{min}\right)^2} & f < f_{avg} \\ \dfrac{P_{m1} + P_{m2}}{2} + \dfrac{\left(P_{m1} - P_{m2}\right)\left(f - f_{avg}\right)^2}{2\times\left(f_{max} - f_{avg}\right)^2} & f \geqslant f_{avg} \end{cases} \tag{13}$$

Where f_{max} is the maximum individual fitness value in the population; f_{min} is the minimum individual fitness value in the population; f_{avg} is the average fitness value of population; f' represents the larger fitness values among two crossing individuals; f represents fitness values of individuals to be mutated; $P_{c1} = 0.9, P_{c2} = 0.4, P_{m1} = 0.1, P_{m2} = 0.005$.

In the early stage of the genetic algorithm, a specified number of individuals were randomly generated within the range of joint parameter movement, and the corresponding individual parameters were substituted into Eq. (11) to obtain the early fitness value. With the continuous operation of the algorithm, the fitness values also increase. Combining Eq. (12) and Eq. (13), the curves of crossover probability and mutation probability with fitness values are obtained as shown in Fig. 3. It is easy to see that the novel P_c and P_m conform to the principles of adjustment.

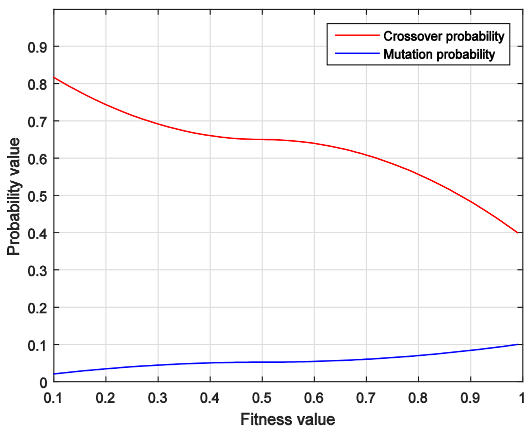


Fig. 3. Cross probability and variation probability curve

7) Choosing immigration operator and artificial selection operator

Immigration operator combined with population substitution and individual substitution is determined to further improve convergence speed and accuracy. In the process of algorithm operation, every population is sorted according to the fitness value to select the best population and the worst population, and replace the worst population to the best population, as shown in Fig. 4. The best and worst individuals of each population are selected by individual evaluation operator, and the worst individuals of the latter population are replaced by the best individuals of the former. The worst individual of the first population is replaced by the best individual of the last population to ensure the algorithm closure. Population substitution must be done before individual substitution to optimize the direction of algorithm evolution.

Artificial selection operator is used to obtain the best individuals of other population and store them in elite populations. To guarantee the best individuals not to be destroyed or lost, the elite population only updates. Finally, individual with the highest fitness is selected from elite populations.

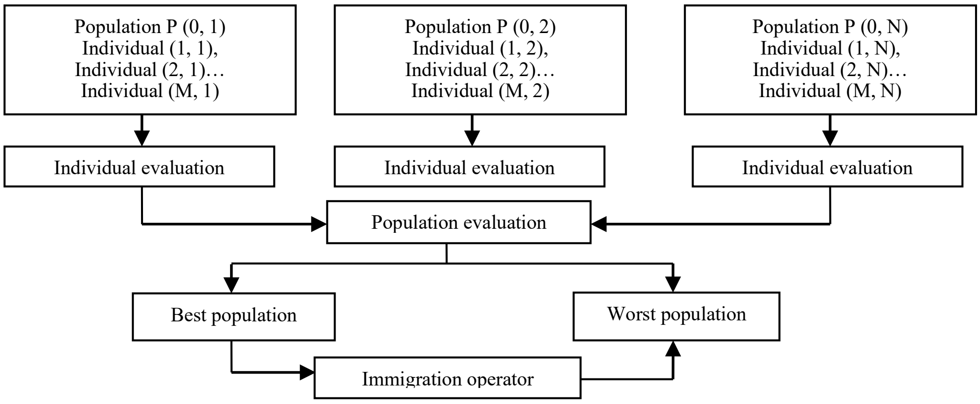


Fig. 4. Population replacement process

4. Simulation analysis of inverse kinematics solution

The improved multi-population adaptive genetic algorithm is applied to a redundant manipulator through simulation, and compared with the TMPGA to show the correctness, stability and convergence of our algorithm. The simulations are implemented with the aid of the MATLAB R2015a.

4.1 Correctness Analysis of Algorithms

The parameters of the two algorithms are set up as follows:

The TMPGA: binary coding, roulette selection, multi-point crossover, T=150, M=100, N=10,

$$P_c^* = 0.4 + (0.9 - 0.4) \times rand(\mathrm{N}, 1) \tag{14}$$

$$P_m^* = 0.005 + (0.1 - 0.005) \times rand(\mathrm{N}, 1) \tag{15}$$

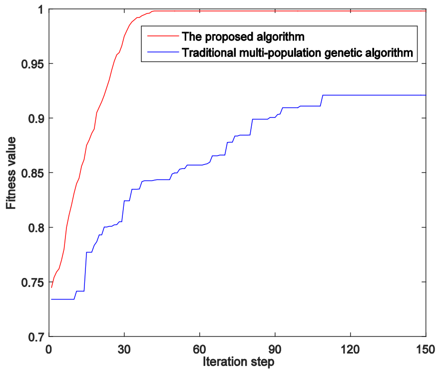
$$Fitness^* = \frac{1}{1 + \mu \cdot f_1(x) + \beta \cdot f_2(x)} \tag{16}$$

Where $\mu = 0.001, \beta = 1$.

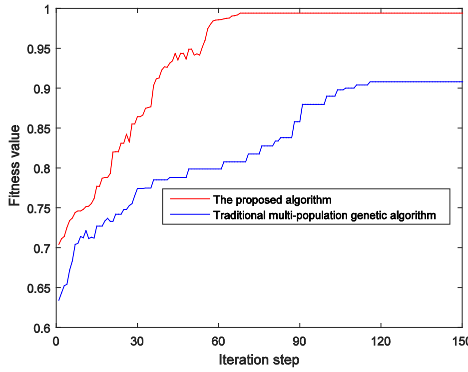
The proposed algorithm: real number coding; roulette selection; multi-point crossover combined with uniform crossover; multi-point crossover combined with uniform crossover; Eq. (11), (12) and (13) are used to determine the fitness function, P_c and P_m, respectively; immigration operator and artificial selection operator are used, T= 150, M=100, N=10,

Two algorithms are used to solve the IK solutions for non-singular configuration **A** and singular configuration **B**, and the corresponding evolution curve of fitness values are shown in Fig. 5.

$$\mathbf{A} = \begin{bmatrix} -0.866 & 0 & 0.5 & -177 \\ 0 & -1 & 0 & 0 \\ 0.5 & 0 & 0.866 & 105 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & -1 & -30 \\ -1 & 0 & 0 & -120 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a)



(b)

Fig. 5. (a) Evolution curve of fitness values of nonsingular configuration **A**, (b) Evolution curve of fitness values of singular configuration **B**.

The IK solutions of **A** and **B** through the proposed algorithm are (20.05; 1.57; -0.523; 0; 0.00016; 0.00028; 0.00031) and (50.06; 0.003; 1.57; 1.57; 1.57; 0; 0), respectively. Also, the IK solutions of **A** and **B** through the TMPGA are (21; 1.62; -0.533; 0; 0.0013; 0.002; 0.0014) and (50.161; 0.003; 1.58; 1.57; 1.57; 0; 0), respectively. According to Eq. (11), the closer of the fitness value is to 1, the higher of solving accuracy is. The optimal fitness values of **A** and **B** by the proposed algorithm are 0.998 and 0.994, which are higher than the optimal fitness values of 0.9215 and 0.9006 by the TMPGA. Therefore, the proposed algorithm is more accurate in solving the IK solutions of redundant manipulators under non-singular and singular configurations than the TMPGA.

4.2 Stability and Convergence Analysis of the Algorithms

In the workspace of the manipulator, sixty discrete points (distributed in non-singular and singular configurations) are randomly selected, and their IK solutions are solved by the two algorithms. The optimal fitness values of these discrete points are obtained, as shown in Fig. 6. The optimal fitness values corresponding to the proposed algorithm are high and the fluctuation is small, i.e., the accurate and the stability of the proposed algorithm are better than the TMPGA.

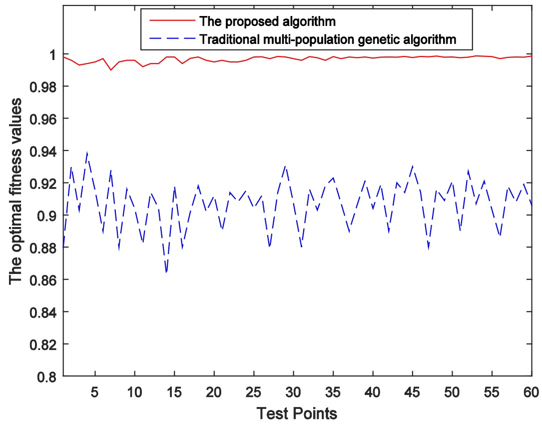


Fig. 6. Fitness function curve of discrete points

Furthermore, using Eq. (7) and (8), the position errors and orientation errors of sixty discrete points by the two algorithms are obtained as shown in Fig.7 and Fig.8. It is not difficult to find that the pose errors obtained by the proposed algorithm are significantly smaller than the TMPGA.

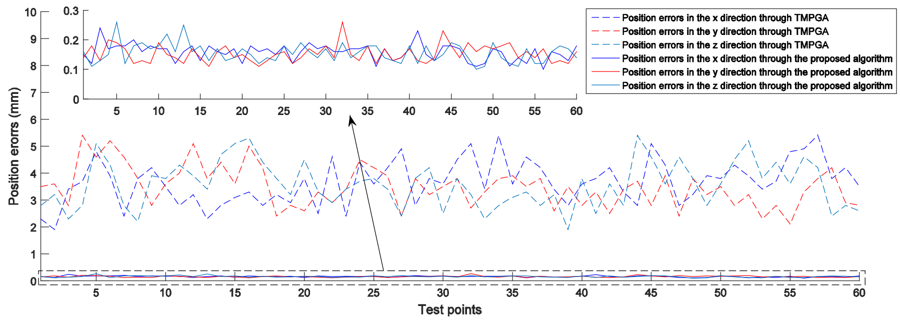


Fig. 7. Position errors in the *x, y, z* direction

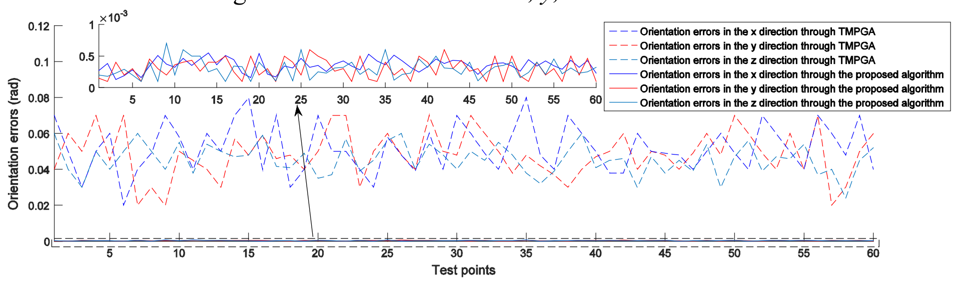


Fig. 8. Orientation errors in the *x, y, z* direction

Table 2

Analysis of test data

Algorithm type	Average position error (mm)	Average orientation error (rad)	Average iteration
TMPGA	3.89	0.051	114
Proposed algorithm	0.15	0.00023	53

The average position error and orientation error calculated by the two algorithms are shown in Table 2. The average error obtained by the TMPGA is 3.89mm, while the average position error obtained by our algorithm is 0.15mm, which is a reduction of 96%, and the average orientation error is reduced from 0.051rad to 0.00023rad, which is a reduction of 99%. The average number of iteration steps by our algorithm is 53, which is faster than the average number of steps of 114 by the TMPGA. Therefore, the improved multi-population adaptive genetic algorithm is significantly higher than the TMPGA in solving accuracy and convergence speed.

5. Conclusion

An improved multi-population adaptive genetic algorithm is proposed to solve the IK solution. Firstly, the inverse solution problem is transformed into an optimization problem, and the objective function and the constraint function are established. Secondly, aiming at the shortcomings of the TMPGA, some improvement measures are put forward. Finally, our algorithm is simulated in a 7-DOF manipulator and compared with the TMPGA. The results show that the pose errors obtained by our algorithm are significantly smaller than the TMPGA, and the convergence speed is also better than the TMPGA.

Acknowledgments

This work was supported by the Major Science and Technology Innovation Projects in Shandong Province, China (2017CXGC0919) and National Natural Science Foundation of China Youth Fund (61803235).

REFERENCES

[1]. *L. Predescu, I. Stroe.* “On the kinematics of the scorbot ER-VII robot”, University Politehnica of Bucharest, Scientific Bulletin, Series D: Mechanical Engineering, **vol. 77**, no. 2, 2015, pp.35-46.

[2]. *B. SICILIANO*, *K. OUSSAMA*. Springer handbook of robotics, Springer Science & Business Media, Berlin, 2008.

[3]. *W. E. SCHIESSER*. The numerical method of lines, Academic Press, California, 1991.

[4]. *W. Hui, Q. Feng*. “A survey of swarm intelligence optimization algorithm”, Control and Instruments in Chemical Industry, **vol. 34**, no. 5, Oct. 2007, pp.7-13.

[5]. *R. Gao*. “Inverse kinematics solution of Robotics based on neural network algorithms”, Journal of Ambient Intelligence and Humanized, **vol. 11**, no. 3, Dec. 2020, pp.6199-6209

[6]. *G. W. Jiang, M. Z. Luo, K. Q. Bai, S. X. Chen.* “A Precise Positioning Method for a Puncture Robot

Based on a PSO-Optimized BP Neural Network Algorithm", Applied Sciences, **vol. 7**, no. 10, Sep. 2017, Article Number: 969.

[7]. *Y. H. Tang, Y. Q. Zhuang, L.P. Shi, Y. G. Jia.* "Kinematic analysis of a Stewart platform based on AFSA", University Politehnica of Bucharest, Scientific Bulletin, Series D: Mechanical Engineering, **vol. 81**, no. 3, 2019, pp.15-26.

[8]. *P. Costa, J. Lima, A. I. Pereira, P. Costa, A. Pinto*. "An Optimization Approach for the Inverse Kinematics of a Highly Redundant Robot", Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015. Advances in Intelligent Systems and Computing, Springer, Cham, **vol. 427**, Jan. 2016, pp.433-442.

[9]. *Z. F. Cai, D. P. Huang, Y. Q. Liu.* "Inverse kinematics of multi-joint robot based on particle swarm optimization algorithm", Journal of Automation and Control Engineering, **vol. 4**, no. 4, Aug. 2016, pp.305-308.

[10]. *Z. W. Ren, Z. H. Wang, L. N. Sun.* "A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-DOF redundant humanoid manipulator", Frontiers of Information Technology and Electronic Engineering, **vol. 16**, no. 7, July. 2015, pp.607-616.

[11]. *G. Pachicano, R. L. Angel, G. Quiroz, L. Torres*. "Inverse kinematics estimation based on smart optimization for biomechanical linkages", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, **vol. 213**, Mar. 2018, pp. 40-48.

[12]. *Z. Riekert, M. Helbig*. "Solving Inverse Kinematics with Vector Evaluated Particle Swarm Optimization", International Conference in Swarm Intelligence, Springer, Cham, **vol. 10385**, June. 2017, pp. 225-237.

[13]. *J. S. Wang, S. X. Li.* "An Improved Grey Wolf Optimizer Based on Differential Evolution and Elimination Mechanism", Scientific Reports, **vol. 9**, no. 1, May. 2019, Article Number: 7181.

[14]. *M. Mitchell.* An introduction to genetic algorithms, MIT Press, Cambridge, 1998.

[15]. *J. H. Holland.* Adaptation in natural and artificial systems, MIT Press, Massachusetts, 1992.

[16]. *K. A. DeJong*. An analysis of the behavior of a class of genetic adaptive systems, University of Michigan Press, Ann Arbor, 1975.

[17]. *M. Lin, G. Wang, Y. C. Lin.* "Robot inverse kinematics based on improved genetic algorithm. Journal of Jiangsu University of Science and Technology", **vol. 26**, no. 4, Nov. 2012, pp. 370-375.

[18]. *Y. Lin, H. Zhao, H. Ding, L. Yang, Z. Huan, D. Han*. "Solution of inverse kinematics for general robot manipulators based on multiple population genetic algorithm", Journal of Mechanical Engineering, **vol. 53**, no. 3, Feb. 2017, pp. 1-8.

[19]. *X. Q. Lv, M. Zhao*. "Application of Improved BQGA in Robot Kinematics Inverse Solution", Journal of Robotics, Feb. 2019, Article ID: 1659180. https://doi.org/10.1155/2019/1659180.

[20]. Y. J. Lei, S. W. Zhang. MATLAB genetic algorithm toolbox and application, Xi'an University of Electronic Science and Technology Press, Xi'an, 2014.

[21]. *M. Alshraideh, L. Tahat.* "Multiple-population genetic algorithm for solving min-max optimization problems", International Review on Computers and Software, **vol. 10**, no. 1, Jan. 2015, pp.9-19.

[22]. *R. B. Agrawal, K. Deb*. "Simulated binary crossover for continuous search space", Complex Systems, **vol. 9**, no. 2, April. 1995, pp. 115-148.

[23]. *F*. Zhu. "Applications and Improved Schemes of Genetic Algorithms in Structural Optimization with Discrete Variables", World Earthquake Engineering, **vol. 18,** no. 4, Dec. 2002, pp. 132-133.

[24]. *Y. Run.* A new adaptive genetic algorithm, Harbin Engineering University, Harbin, 2007.

[25]. *M. Li.* Improvement of genetic algorithm and application in optimization problems, Jilin University, Changchun, 2006.

[26]. *G. L. Cheng, X. F. Wang*. Genetic Algorithms and Applications, People's Posts and Telecommunications Press, Beijing, 1996.

[27]. *M. Srinivas, L. M. Patnaik*. "Adaptive probabilities of crossover and mutation in genetic algorithms", IEEE Transactions on Systems, Man and Cybernetics, **vol. 24**, no. 4, Apr. 1994, pp.656-667.

[28]. *Z. W. Ren*, Z. San. "Improved Adaptive Genetic Algorithm and its Application Research in Parameter Identification", Journal of Systems Simulation, **vol. 18**, no. 1, Jan. 2006, pp. 41-66.