2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Interval Search Genetic Algorithm Based on Trajectory to Solve Inverse Kinematics of Redundant Manipulators and Its Application

Di Wu, Wenting Zhang, Mi Qin and Bin Xie*

Abstract— **In this paper, a new method is proposed to solve the inverse kinematics problem of redundant manipulators. This method demonstrates superior performance on continuous motion by combining interval search genetic algorithm based on trajectory which we propose with parametric joint angle method. In this method, population continuity strategy is utilized to improve search speed and reduce evolutionary generation, interval search strategy is introduced to enhance the search ability and overcome the influence of singularity, and reference point strategy is used to avoid sudden changes of joint variables. By introducing those three strategies, this method is especially suitable for redundant manipulators that perform continuous motion. It can not only obtain solutions of inverse kinematics quickly, but also ensure the motion continuity of manipulator and accuracy of the end effector. Moreover, this algorithm can also perform multi-objective tasks by adjusting the fitness function. Finally, this algorithm is applied to an 8 degree of freedom tunnel shotcrete robot. Field experiments and data analysis show that the algorithm can solve the problem quickly in industrial field, and ensure the motion continuity and accuracy.**

I. INTRODUCTION

In recent years, intelligent robots are attracting more and more attention [1]. The widely application of intelligent robots can not only reduce the threats for workers, but also improve the production efficiency. However, many industrial robots have redundant structures or do not meet the Pieper criterion [2], so it is impossible to calculate the analytic solutions of inverse kinematics directly.

Inverse kinematics means that all joint variables of the manipulator can be obtained for a specific position and orientation of the end effector. This process is not trivial since it involves multivariable transcendental equations. Generally, only the manipulator meeting Pieper criterion is considered to be able to obtain analytic solutions. Moreover, if the manipulator exceeds six degree of freedom (DOF), redundancy is considered. In this situation, it is invalid to obtain joint variables only by solving equations. Generally, there are infinite soultions for redundant manipulators. Typically, inverse kinematics of redundant manipulators (IKRM) can be solved from two aspects, respectively velocity and position.

Since 1970s, algorithms based on velocity have been used to solve IKRM problem. S. Lee [3] et al. proposed the joint decomposition method using parametric joint method combined with gradient descent method. A. Abdelrahem [4] et al. proposed a new method to solve the problem of rigid constraints of manipulator joints. However, these methods only transform the hard constraint problem into the soft constraint problem. This method can not fully guarantee the validity of solutions and also can not fully utilize the full reachable space of the manipulator. M. Shimizu [5] et al. carried out structural analysis of a 7-DOF manipulator, whose joints are all rotating joints, and then obtained all feasible solutions. However, this method can only be applied to some special manipulators and can not be widely applied. In [6], a novel iterative algorithm for IKRM was proposed and it has superior performance on avoiding obstacles. Nevertheless, it cost too much time to choose the optimal configuration.

In recent years, IKRM problem is usually regarded as a constrained optimization problem [7][8]. Thus, IKRM algorithms based on position level have received more attention. Many strategies, mainly including evolutionary algorithm and learning model, can be applied quickly for IKRM problem. Some researchers proposed several approaches to approximate the inverse kinematic model, including SVM [9][10], work-area decomposition and matching [11], symbolic regression [12], etc. Soutions can be generated very quickly and repeatedly if the inverse model is learned accuracyly. However, for a specified target point, multiple solutions may exist. The entire solutions are difficult to obtain by those methods, which will reduce the workspace and flexibility of manipulators. Besides, retraining process is necessary if the kinematic geometry changes or any changes in objectives and constraints. Thus those learning algorithms are difficult to apply for dynamic tasks.

Evolutionary algorithms are extraordinary to deal with IKRM in robust and flexibility. This framework can be expanded to any manipulators easily. S. Momani [13] et al. applied continuous genetic algorithm to a 3-DOF manipulator. The motion curves of all joints are very smooth. But the performance of this algorithm will be poorer with the increase of robot's DOF. In [14], M. Ahmad et al. proposed a hybridization algorithm which combined general genetic algorithm (sGA) and the Newton-Raphson method to solve IKRM problem. In [15], evolutionary algorithm is combined with gradient-based method to solve the kinematics problems of generic full-body robot. This method can search the solutions of robots quickly. Besides, swarm algorithm [16][17] and firefly algorithm [18][19] are also applied to IKRM problems successfully.

However, there are several problems in solving inverse kinematics based on evolutionary algorithm.

1) All joints are set as parameters, and the search space is huge.

Di Wu, Wenting Zhang, Mi Qin and Bin Xie are with School of Automation, Central South University, Changsha, China. Bin Xie is also with Hunan Xiangjiang Artificial Intelligence Academy, Changsha, China. Bin Xie is the corresponding author, email:xiebin@csu.edu.cn

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

2) The accuracy of end-effector is the primary task. Other tasks, such as ensuring joint continuity and avoiding joint limitation, can not be well guaranteed.

3) Sudden change happens easily near the kinematic singularity.

Thus, in this paper, parametric joint angle method is used to solve the problem 1 and we proposed a new improved genetic algorithm to solve the problem 3. The problem 2 can be solved by combining those two methods.

The core contributions of the paper include: 1) A new idea for solving IKRM problem based on trajectory. 2)A robust and fast improved genetic algorithm, interval search genetic algorithm based on trajectory(ISGABT), which contains three strategies, population continuity, interval search and reference point strategy. 3) A general method combined parametric joint angle method with ISGABT to solve inverse kinemtaics problem of redundant manipulators which perform continous motion tasks.

II. ALGORITHMIC APPROACH

The combination of parametric joint method and ISGABT can solve the IKRM problem of continuous motion effectively. The framework of this method is shown in Fig. 1.

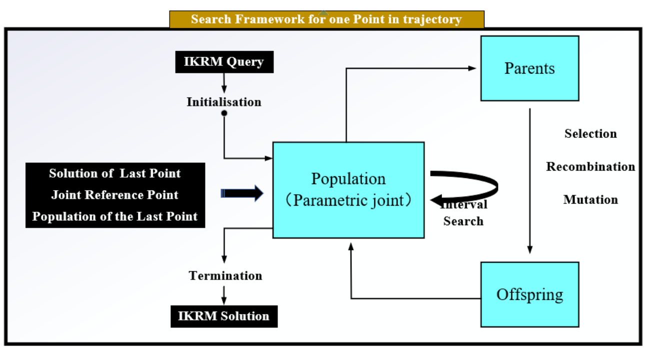

Fig. 1: Search Framework for One Point in Trajectory

Firstly, the structure of the manipulator is analyzed. And then those joints, especially whose motion characteristics are obvious, are selected as parametric joints. Thereafter, the parameterized joints are selected as variables in ISGABT. Finally, by setting reasonable objective functions, solutions meeting the task requirement can be obtained. In order to ensure that the algorithm can search the optimal solution globally in a short time, this paper improves the genetic algorithm in three aspects.

- Population Continuity Strategy. In one trajectory, the next point inherits the final population of the previous point. By this, not only the motion continuity is guaranteed, but also the evolution of population is accelerated and the convergence time is reduced.
- Interval Search Strategy. For manipulators which perform continuous trajectory motion, generally optimal solutions are near the previous solutions. This strategy searches solutions only near the previous solutions. Therefore, the continuity of manipulator can be guaranteed and solutions can be searched quickly. But if joint value exceeds the joint limitation or solutions are absent, the manipulator must change its robot posture. In this case, to ensure the robustness of the system, this strategy searches for neighborhood range firstly and then enlarges the search range. Moreover, although there are infinite solutions for IKRM, the effective solutions decrease sharply near the kinematic singularities. By using this strategy, the search ability enhances dramatically and thus the sudden changes of joint variables can be avoided.
- Reference Point Strategy. Above two strategies can lead to a problem, i.e., the value of parameterized joints tends to be fixed or change little. In this case, in order to reach the target point, the other joints will change larger, and the value of non-parametric joints is easy to reach the joint limitation. The posture of manipulator has to change suddenly and the orientation of the end effector may be uncontrollable for a short time. Therefore, it is necessary to set reference points to guide the motion of some joints.

A. *Parametric joint angle method*

Parameterized joint angle method greatly reduces the search space, and makes all solutions meet the accuracy requirements. Usually the relationship between target matrix X and n-DOF joint variables θ_i can be expressed as (1).

$$X = f(\Theta_1, \Theta_2, ..., \Theta_n) \tag{1}$$

Tasks need m-DOF to complete ($n > m$). Therefore, the manipulator has $n - m$ dimension redundancy. The parameterized joint angle method needs to select $n - m$ redundant joints to parameterize. This operation transforms a n-DOF redundant manipulator into a m-DOF non-redundant manipulator, while the remaining $n-m$ parameterized joints are used for optimization. Joint space is also divided into dimension reduction space and parameter space. The joint variables in dimension reduction space are represented by Θ_r, while that in parameter space are represented by Θ_p. Therefore, the target matrix X is the function of Θ_r and Θ_p, as shown in (2).

$$X = f(\Theta_r, \Theta_p) \tag{2}$$

When Θ_p and X are obtained, the joint variables in dimension reduction space can be calculated as (3).

$$\Theta_r = f^{-1}(X, \Theta_p) \tag{3}$$

Thus, the closed analytic solution of non-redundant joint can be obtained.

B. *ISGABT*

ISGABT solves IKRM problem based on trajectory. The population and solutions of the previous point are inputted to ensure the continuity, real-time and robustness. Its algorithm framework is shown in Algorithm 1.

Functions, $Inverseanaly$ and $ForwardKine$ in Algorithm 1, are closed analytical solution of non-redundant joints and forward kinematics of manipulator respectively.

Algorithm 1: ISGABT

Input : Population Size N, Target Matrix X
Output: Optimal solutions

1 **Initialize** Population P
2 **for** *each trajectory* **do**
3   **Generate** reference points
4   **Create** initial desired posture
5   **for** *each point* **do**
6   **Inherit** population from the previous point
7   **Reset** the values of reference joints
8   **Determine** search range
9   **for** $i \leftarrow 1$ *to* 10000 **do**
10       $\mathbf{V} \leftarrow Binarytoreal(P, S, \Theta_C)$
11   **for** *each individual* **do**
12         $Solution \leftarrow Inverseanaly(V, X)$
13         $X_c \leftarrow ForwardKine(Solution)$
14   **if** $X_c - X = 0 \bigcap$ *Satisfy all joint limitation* **then**
15   **Calculate** fitness
16   **if** $min(fitness) < threshold$ **then**
17   **Preserve** population and optimal solution
18   **Break**
19   **else**
20   **Selection**
21   **Recombination**
22   **Mutation**
23   **Create** offspring

1) Joint Variable Encoding: In this paper, binary coding is used to encode parametric joints corresponding to line 1 in algorithm 1. For $n-m$ parameters, the corresponding code Bin is set in (4).

$$Bin = (B_{l_1}|B_{l_2}|...|B_{l_{n-m}}) \tag{4}$$

Binary coding of each joint is denoted as B_{l_i} and l_i represents the binary length of each parameter, $i = 1, ..., n-m$. The length of coding determines the resolution power of parameters and the size of search space.

2) Reference points generating: The reference points are set for joint with obvious motion characteristics corresponding to line 3 in algorithm 1. For joint i, initial joint value, terminal joint value and number of point in a trajectory are denoted as V_s, V_t and N_t respectively. The interval sp can be calaulated by (5) and the jth reference point R_j can be obtained from (6).

$$sp = (V_s - V_t)/(N_t - 1) \tag{5}$$

$$R_j = V_s - (j-1) * sp, j = 1, 2, ..., N_t \tag{6}$$

Because of the existence of search interval strategy, the reference vector does not need to be completely accurate.

3) Initial desired posture creating: This part is corresponed with line 4 in algorithm 1. The previous solutions are required to calculate the fitness. However, the first point in trajectory does not have previous point. Generally, this initial soltion can be set as zero vector.

4) Values of reference joints resetting: This part is corresponed with line 7 in algorithm 1. Resetting the values of reference joints as R_j in θ_c. For some common trajectories, such as lines and semicircles, this method can accelerate the convergence of the algorithm. However, it is not applicable for joints which are difficult to predict their motion and under this condition, we do not need to reset the value of reference joints.

5) Search interval determining: Specific process is shown in algorithm 2.

Algorithm 2: Determine search range

Input : Population P, Target Matrix X, Previous Solutions Θ_C, Search Radius Rad
Output: Search Interval S

S $\leftarrow Rad$
for $i \leftarrow 1$ *to* 70 **do**
  $Count \leftarrow 0$
  $\mathbf{V} \leftarrow Binarytoreal(P, S, \Theta_C)$
  if $i > 5 \bigcap i \leq 15$ **then**
    $S \leftarrow 2 * Rad$
  else if $i > 15 \bigcap i \leq 30$ **then**
    $S \leftarrow 4 * Rad$
  else if $i > 30 \bigcap i \leq 50$ **then**
    $S \leftarrow 8 * Rad$
  else
    $S_{max} \leftarrow Maximum\ Search\ Range$
  for *each individual* **do**
    $Solution \leftarrow Inverseanaly(V, X)$
    $X_c \leftarrow ForwardKine(Solution)$
    if $X_c - X = 0 \bigcap$ *Satisfy all joint limitation* **then**
      $Count \leftarrow Count + 1$
  if $Count > 0$ **then**
    Exit
  else
    Initialize Population

6) Binary to real number in search interval: The formula for binary to real number V is shown in (7) corresponding to line 10 in algorithm 1.

$$V = (\sum_{i=0}^{L} 2^i * P_i) * 2 * S/2_L - S + \Theta_C \tag{7}$$

Among them, L denotes the length of binary coding.

7) Fitness calculating: This part is corresponed with line 15 in algorithm 1. Several fitness functions are listed in this paper. In (8), $disC$ is the function of continuity. In (9), $disL$

is the function of joint limitation. And in (10), $disD$ is the function of avoiding obstacles.

$$disC = \frac{1}{1 - e^{-(\frac{x-\Theta_c}{fiaC})^2}} \tag{8}$$

$$disL = \frac{1}{1 + e^{\frac{x-\Theta_{min}}{fiaL}}} + \frac{1}{1 + e^{-\frac{x-\Theta_{max}}{fiaL}}} \tag{9}$$

$$disD = \begin{cases} 0 & if\ d - d_{min} > 0 \\ \frac{1}{d_{min}-d} & else \end{cases} \tag{10}$$

where Θ_c denotes previous solutions, Θ_{min} and Θ_{max} represent the minimum and maximum of joint respectively, d_{min} is the shortest distance from an obstacle, $fiaC$ and $fiaL$ are scaling ratio.

While Θ_c, $fiaC$, Θ_{min}, Θ_{max}, $fiaL$, and d_{min} are set as 0, 2, -8, 8, 0.05, 5 respectively. Variation curves are shown in Fig. 2.

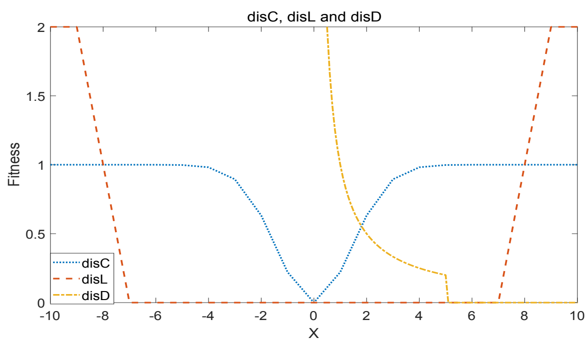

Fig. 2: Fitness Function

The image of $disC$ is similar to a basin. When the new solution closes to the previous solution, the value of fitness is small. This value can sharply increase when the distance between the new solution and the previous one increases slowly. By modifying $fiaC$, the intensity of increasement will change. Different weight and $fiaC$ can be assigned for each joint to distribute the priority of joint continuity. In Fig. 2, $disL$ is the function to keep all joints far away from the joint limitation. When solution closes to -8 or 8, the fitness rises quickly. This function can ensure the large range of joint movement space and avoid joint reaching limtation. And $disD$ is the function to avoid obstacles. When the distance is less than d_{min}, similarly, the fitness will increase sharply.

8) Selection: Roulette and elite strategy are combined to select offsprings from parents. Several best individuals are chosen from parents and others will be chosen by roulette.

9) Recombination: The coding of each variable operates independently multi-point crossover.

10) Mutation: The coding of each variable operates independently multi-point mutation.

III. Algorithm Application

In this paper, the parametric joint method and ISGABT were combined to apply to the 8-DOF tunnel shotcrete robot KC-30. This robot will be used to perform automatic grouting operations under tunnels. The end nozzle of the manipulator must be perpendicular to the tunnel wall and ensure the continuity of the trajectory during the movement. If joint movement is incoherent or joint variables change suddenly, the end orientation may lose control, resulting in the random spraying of concrete and causing waste and insecurity. Kinematic modeling of manipulator is the basis of trajectory planning and real-time control. The forward kinematic model of the manipulator must be accurate, and the inverse kinematic solutions can be obtained quickly and ensure the continuity of trajectory.

A. Forward Kinematics

The KC-30 robot has eight joints, of which the 3rd and 6th joints are expansion joints and the other joints are revolving joints. In this paper, kinematics model of 8-DOF manipulator is built based on D-H method. The reference coordinates of all joints are shown in Fig 3.

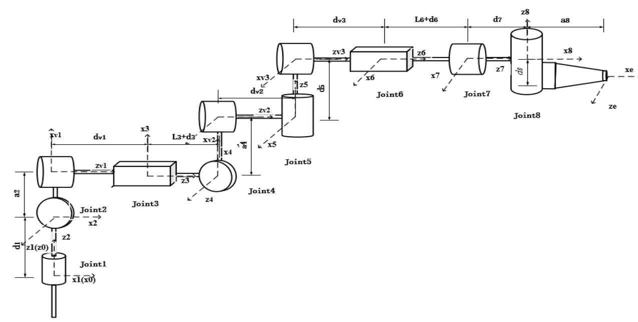

Fig. 3: Structural Sketch of manipulator

Three virtual joints are designed in the model of forward kinematics to ensure that model is bulit along fuselage. Thus the measurements required by the model can be obtained from the CAD drawings. Based on Fig.3, D-H parameter table is shown in table 1.

TABLE I: D-H Parameter Table

Joint	Θ_i	$d_i(mm)$	$a_i(mm)$	α_i	Offset
1	Θ_1	770	0	pi/2	0
2	Θ_2	0	557	pi/2	pi/2
Virtual 1	0	2670	0	0	0
3	0	L_3+1030	0	-pi/2	0
4	Θ_4	0	485	pi/2	0
Virtual 2	0	1800	0	pi/2	pi/2
5	Θ_5	1425	0	-pi/2	0
Virtual 3	0	16.359	0	0	0
6	0	L_6+2693.7	0	0	0
7	Θ_7	663	0	pi/2	0
8	Θ_8	1138	464	pi/2	pi/2

The coordinate transformation matrix between coordinate

i and coordinate $i-1$ is shown in (11).

$$A_i^{i-1}(q_i) = \begin{bmatrix} C_{\Theta_1} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$sin(\theta_i)$ is denoted as S_{θ_i} and $cos(\theta_i)$ is denoted as C_{θ_i}. The target matrix X can be obtained by (12).

$$X = A_1^0(q_1)A_2^1(q_2)...A_n^{n-1}(q_n) \tag{12}$$

B. Parametric joint angle

The KC-30 robot has eight joints and two joints are redundant. Revolving joint 1 and expansion joint 6 were selected as parametric joints. Joint 1 is a rotating joint. When a robot sprays, the joint changes from right to left or from left to right. Therefore, the motion characteristic of joint 1 is obvious. And joint 6 was chosen as parametric joint for solving the analytical solution of non-redundant joint easier. The objective matrix can be described as (13).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Therefore, the function of inverse kinematics can be obtained by $T_n^0(q) = R$. Due to the limitation of the length of the article, this paper only lists the analytical solution of joint 5, and the solution of other joints adopts the same method.

θ_1 was separated from other joints according to (14).

$$A_1^0(q_1)^{-1}T_n^0(q) = A_2^1(q_2)...A_n^{n-1}(q_n) \tag{14}$$

12 equations can be obtained by (14), four equations in the second line of (14) are shown in (15).

$$\begin{cases} r_{21}C_1 + r_{11}S_1 = C_8S_5 - C_5C_7S_8 \\ r_{22}C_1 + r_{12}S_1 = S_5S_8 + C_5C_7C_8 \\ r_{23}C_1 + r_{13}S_1 = -C_5S_7 \\ P_yC_1 + P_xS_1 = d_8(C_8S_5 - C_5C_7C_8) \\ \qquad +S_5(L_6 + d_6) + d_7S_5 \\ \qquad -a_8C_5S_7 \end{cases} \tag{15}$$

The first and third equations in (15) can be brought to the fourth equation, and then equation (16) can be gotten.

$$\begin{aligned} &(P_y - d_8r_{21} - a_8r_{23})C_{\theta_1} + (P_x - \\ &d_8r_{11} - a_8r_{13})S_{\theta_1} = (L_6 + d_6 + d_7)S_{\theta_5} \end{aligned} \tag{16}$$

The analytical solution of θ_5 can be obtained as (17).

$$\theta_5 = arcsin(\frac{AC_{\theta_1} + BS_{\theta_1}}{L_6 + d_6 + d_7}) \tag{17}$$

$$A = P_y - d_8r_{21} - a_8r_{23}, B = P_x - d_8r_{11} - a_8r_{13}$$

Similarly, the analytic solution of θ_2, L_3, θ_4, θ_7, θ_8 can be achieved by this approach.

C. Solving IKRM Problem Based on ISGABT

The joint variables, θ_1 and L_6, were set as parameters in ISGABT. Besides, the expected trajectory was obtained by scanning the tunnel on the field. In the process of movement, the end effector is expected to be vertical to the tunnel and the expected orientation of the end effecotr is designed by XZX method in Euler angle. Thus the value of the second x (denoted as x_0) is not solid. The coding length of θ_1, θ_6 and x_0 are 11, 11 and 9 respectively. And the range of θ_1, θ_6 and x_0 are [-30,30]°, [0,2000]mm and [-90,90]° respectively. The resolution power of them are 0.029°, 0.97mm and 0.35°, respectively. The actual tunnel scanning data are shown in Fig. 4.

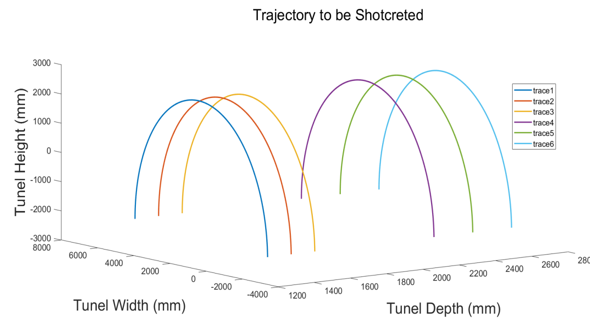

Fig. 4: Trajectory to be shotcreted

Every trace was constituted with 129 points. The reference points of θ_1 range from 17° to $-17°$ and its search radius was 1°. The reference points of L_6 and x_0 were not used and their search radius were 100mm and 5° respectively. The population was set at 350. The probability of crossover and mutation was 0.7 and 0.1, respectively. Our goals are to ensure the joint continuity and avoid joint limitation. The weight vector of disC for all joints was [0.1,0.2,0.2,0.1,0.1,0.1,0.1,0.1] with fiaC vector was [5,1,520,2,2,520,1,1]. And the weight vector of joint limitation for all joints was [0.3,0.3,0,0.1,0.1,0,0.1,0.1] with fiaL vector was [0.8,0.6,25,0.6,0.8,25,0.8,1]. The weight vector of continuity and joint limitation was [0.6,0.4].

Solutions for tunnel trace by SGA and ISGABT can be obtained. The comparsion between SGA and ISGABT is shown in Table II.

TABLE II: Comparsion between SGA and ISGABT

Algorithm	Average Evolution Generation	Average Time(s)	Success Rate(%)
SGA	29.99	1.866	58.91
ISGABT	1.137	0.0192	100

All solutions of IKRM by ISGABT can be obtained after 14.86s and the average time of every point is 19.2ms. SGA spent 1444.284s to solve 774 points and the average time of every point is 1.82s. Obviously, ISGABT can meet the requirement of real time operation in industrial field.

Most points can find the optimal solution directly in the first generation by ISGABT, because the population can be inherited in the whole trajectory and the interval search strategy enhance the local search ability. Obviously,

(a) Position 1 (b) Position 2 (c) Position 3 (d) Position 4

Fig. 5: Actual Trajectory Spraying based on Kinematics

SGA without those strategies need far more generations to evolve. Besides, the success rate of ISGABT is 100% while some points are difficult to solve for SGA. In Fig.6, the evolution generation of 1st trace by SGA and ISGABT is demonstrated. Those points in the middle of the trace is more difficult for SGA to search due to the existence of kinematic singularity which leads to feasible solutions decrease sharply. However, the impact of kinematic singularity on ISGABT is little. The optimal solutions are still searched quickly and meet the requirement of tasks.

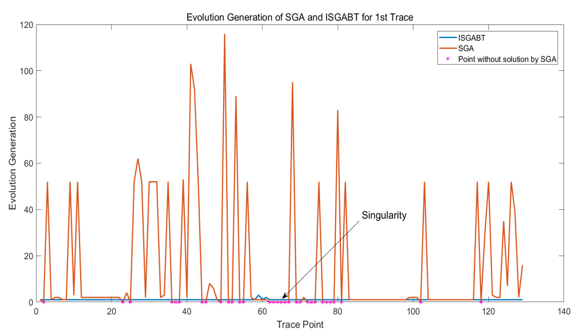

Fig. 6: Evolution Generation of 1st Trace by SGA and ISGABT

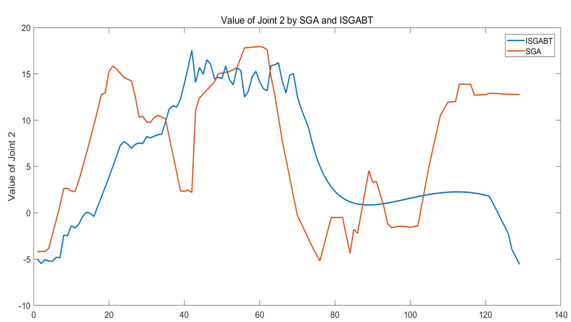

Fig. 7: Value of Joint 2 by SGA and ISGABT

Values of revolving joint 2 and expansion joint 6 by SGA and ISGABT are shown in Fig. 7-8. Solutions solved by ISGABT are more smooth and the difference between two adjacent points is small. Thus the trajectory of the manipulator can be smoother and more coherent by using ISGABT.

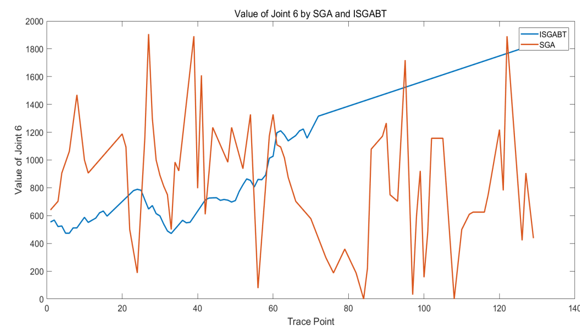

Fig. 8: Value of Joint 6 by SGA and ISGABT

D. Grouting in Tunnel

The actual grouting operation was carried out in the actual tunnel. As shown in Fig. 5, the joint movement is coherent and this algorithm can meet the real-time requirements. In addition, the orientation of the end effector is perpendicular to the tunnel wall, and the position is corresponding with expected position, indicating that the kinematic model is accurate. The feasibility of the algorithm is proved.

IV. CONCLUSIONS

This paper proposes a new mehtod to solve IKRM problem. By using the parametric joint angle method, the joint space of the manipulator is divided into reduced dimension space and redundant space, thus the inverse kinematics problem of the redundant manipulator is transformed into the inverse kinematics problem of the non-redundant manipulator and the constraint optimization problem. The usage of parameterized joint angle method ensures the accuracy of the end effector and ISGABT is effective in solving the continuous trajectory. It not only guarantees the ability of global fast solution, but also ensures the continuity of the joint. Even near the singularity, there will be no sudden change of joints. In addition, the secondary task can be added by adjusting the fitness function. The algorithm has strong portability and can be applied to other manipulators that perform continuous trajectory tasks. Finally, the algorithm is applied to the 8-DOF tunnel shotcrete robot. The experimental results show that ISGABT can meet requirements of real-time operation, the continuity of trajectory and the accuracy of the end effector in the industrial field, which proves the feasibility of the algorithm.

REFERENCES

[1] T. M. Wang, Y. Tao, and H. Liu, "Current researches and future development trend of intelligent robot: A review," *International Journal of Automation and Computing*, vol. 15, no. 5, pp. 525–546, 2018.

[2] Pieper and D. Lee, "The kinematics of manipulators under computer control," *Stanford Artificial Intelligence Project*, no. AI-72, 1968.

[3] S. Lee and A. K. Bejczy, "Redundant arm kinematic control based on parameterization," in *IEEE International Conference on Robotics and Automation*, vol. 1, (Sacramento, CA, USA), pp. 458–465, 1991.

[4] A. Abdelrahem, P. Dimitrios, and D. Zoe, "Kinematic control of redundant robots with guaranteed joint limit avoidance," *Robotics and Autonomous Systems*, vol. 79, pp. 122 – 131, 2016.

[5] M. Shimizu, H. Kakuya, W. Yoon, and K. Kitagaki, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, 2008.

[6] L. Zhao, J. Zhao, H. Liu, and D. Manocha, "Efficient inverse kinematics for redundant manipulators with collision avoidance in dynamic scenes," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2502–2507, Dec 2018.

[7] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Predictive inverse kinematics for redundant manipulators with task scaling and kinematic constraints," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 278–285, 2019.

[8] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "Opensot: A whole-body control library for the compliant humanoid robot coman," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6248–6253, 2015.

[9] A. Morell, M. Tarokh, and L. Acosta, "Inverse kinematics solutions for serial robots using support vector regression," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4203–4208, 2013.

[10] A. Mustafa, C. Tyagi, and N. K. Verma, "Inverse kinematics evaluation for robotic manipulator using support vector regression and kohonen self organizing map," in *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pp. 375–380, Dec 2016.

[11] M. Tarokh and M. Kim, "Inverse kinematics of 7-dof robots and limbs by decomposition and approximation," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 595–600, 2007.

[12] F. Chapelle and P. Bidaud, "A closed form for inverse kinematics approximation of general 6r manipulators using genetic programming," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, pp. 3364–3369 vol.4, 2001.

[13] S. Momani, Z. S. Abo-Hammour, and O. M. Alsmadi, "Solution of inverse kinematics problem using genetic algorithms," *Applied Mathematics and Information Sciences*, vol. 10, no. 1, pp. 1–9, 2015.

[14] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 406–420, 2019.

[15] M. Ahmad, N. Kumar, and R. Kumari, "A hybrid genetic algorithm approach to solve inverse kinematics of a mechanical manipulator," *International Journal of Scientific and Technology Research*, vol. 8, pp. 1777–1782, 2019.

[16] S. Dereli and R. Koker, "A meta-heuristic proposal for inverse kinematics solution of 7-dof serial robotic manipulator: Quantum behaved particle swarm algorithm," *Artificial Intelligence Review*, no. 53, pp. 949–964, 2019.

[17] S. Dereli and R. Koker, "Iw-pso approach to the inverse linematics problem solution of a 7-dof serial robot manipulator," *Sigma*, no. 36, pp. 77–85, 2018.

[18] N. Rokbani, A. Casals, and A. M. Alimi, *IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm*, pp. 369–395. Cham: Springer International Publishing, 2015.

[19] D. Serkan and K. Rait, "Calculation of the inverse kinematics solution of the 7-dof redundant robot manipulator by the firefly algorithm and statistical analysis of the results in terms of speed and accuracy," *Inverse Problems in Science and Engineering*, pp. 1–13, 2019.