A Simulation Study on Robot Zero Angle Calibration Based on Improved Firefly Algorithm

Cai Zefan [1, 2], Huang Daoping [1*]

1. Automation College of South China University of Technology
Guangzhou 510640, China;

2. *Department Electronic and Information Engineering,* Shunde Polytechnic, 528300, China

***Abstract* — Absolute accuracy and repeated positioning accuracy are two aspects of the robot. In general, the repeated positioning accuracy is high enough, while the absolute accuracy must be calibrated in order to reach a high level. The robot absolute accuracy is affected by the positioning error, while the positioning error is controlled by many factors, such as manufacturing precision, installation process, sensor error, kinematic parameters, etc.. This paper tries to study how to calibrate the zero position angle error based on the improved firefly algorithm assumed that other factors are accurate. The simulation results show that the method is simple and accurate, and can be applied to the actual robot calibration.**

Keywords - Simulation, robot , software

I. INTRODUCTION

Absolute accuracy and repeated positioning accuracy are two aspects of the robot accuracy. Absolute accuracy is the ability of the robot to reach the set points in the working space, while repeated positioning accuracy stands for the ability of the robot to reach the teaching points again. Typically, the repeated positioning accuracy is high enough, and can reach the level of 0.1mm ~ 0.01mm. Absolute accuracy is not high enough before calibration, just can reach the level of 0.1cm ~ 1cm. After calibration absolute accuracy can be greatly improved and can be nearly equal to the repeated positioning accuracy. The robot absolute accuracy depends on the error of positioning, while the positioning error is controlled by many factors, such as manufacturing precision, installation process, sensor error, kinematic parameters, etc.. [1]

There are two kinds of methods to improve absolute accuracy: [2]

(1) One kind is to improve the robot hardware, such as improving manufacturing accuracy of mechanical components, assembly and installation precision, stiffness of the robot joints, etc.. This kind will increase the cost of manufacturing, and professional workers must be employed in order to guarantee the assembly and the installation precision. This method can't deal with the post positioning error.

(2) The other is to bring in calibration technology of software. In this kind, the robot's actual parameters are identified to improve the structure model of robot. It avoids pouring money and people. When the robot changes, its parameters can be identified again. The calibration technology has a very wide application prospect in improving the absolute accuracy of the robot, and has great research value.

There are a lot of people to research on the calibration technology. Several algorithms are introduced, such as Leveenberg-Marquarde [3], least square [4], maximum likelihood estimation [5], genetic algorithm [6], simulated annealing algorithm [7], etc.. This paper tries to study how to calibrate the zero angle error based on another algorithm the cuckoo search algorithm assumed that other factors are accurate.

Firefly algorithm (FA) [8, 9] is a heuristic algorithm which is developed in 2008 by Xin-She Yang. The brightness of the firefly is a signal system which will attract other fireflies. Because FA is simple and high efficient, it has caused the attention of scholars, and hundreds of papers about FA have been published. FA has been applied successfully in many fields. However, there are no papers about application of robot calibration based on CS. In this paper, Gauss variation and simulated annealing mechanism are introduced to improve the standard FA algorithm. And then, this paper tries to study how to calibrate the zero angle error based on the improve firefly algorithm, and achieves the desired results.

II. D-H PARAMETERS EXPRESSION OF MULTI-JOINT ROBOT

There are several robot pose expressions, such as Euler angle, rotation matrix and four element method, etc.. The most common one is the rotation matrix, and D-H parameters expression is the most famous one of the rotation matrix method. D-H is developed by Denavita and Hartenberg in 1995. [10]

In D-H, the D-H parameters of the link i must be determined firstly. The 4 parameters are link length a_i , link twist a_i , link offset d_i and joint angle θ_i . Then homogeneous coordinate transformation matrix T_i will be calculated with D-H parameters. T_i is a primary coordinate transformation matrix, which describes the relative translation and rotation between the link coordinate systems. T_i is as follow:

$$T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

For rotating joints, a_i, α_i and d_i are constant, while θ_i is variable. Instead for sliding joints, a_i, α_i and θ_i are constant, while d_i is variable.

For a 6 links robot, the terminal pose matrix T is a relative translation between the hand (the 6th joint) and base coordinate system. T is as follow: [10]

$$T = T_1 T_2 T_3 T_4 T_5 T_6. \tag{2}$$

Assuming the position vector of the hand coordinate origin in the base coordinate is p, while the direction vectors are n、o and a, T can be described by a 4×4 matrix as follow:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Fig. (1) is the 3D simulation diagram of a low-cost 6 joints robot, and Fig. (2) is a schematic diagram of the robot and the corresponding coordinate systems. The pose matrix T of the robot is shown as Eq. 3. p、n、o and a in this equation are as follow:

$$\begin{aligned}
p_x &= c_1\left[-c_{23}\left(d_4c_4s_5 - d_5s_4\right) + s_{23}\left(d_6c_5 + d_4\right) + a_1 + a_2c_2\right] \\
&\quad - s_1\left[d_6s_4s_5 + d_5c_4\right], \\
p_y &= s_1\left[-c_{23}\left(d_4c_4s_5 - d_5s_4\right) + s_{23}\left(d_6c_5 + d_4\right) + a_1 + a_2c_2\right] \\
&\quad + c_1\left[d_6s_4s_5 + d_5c_4\right], \\
p_z &= -s_{23}\left(d_4c_4s_5 - d_5s_4\right) - c_{23}\left(d_6c_5 + d_4\right) + a_2s_2, \\
n_x &= c_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) + s_{23}s_5c_6\right] + s_1\left(s_4c_5c_6 + c_4s_6\right), \\
n_y &= s_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) + s_{23}s_5c_6\right] - c_1\left(s_4c_5c_6 + c_4s_6\right), \\
n_z &= s_{23}\left(c_4c_5c_6 - s_4s_6\right) - c_{23}s_5c_6, \\
o_x &= -c_1\left[c_{23}\left(c_4c_5s_6 + s_4c_6\right) + s_{23}s_5s_6\right] - s_1\left(s_4c_5s_6 - c_4c_6\right), \\
o_y &= -s_1\left[c_{23}\left(c_4c_5s_6 + s_4c_6\right) + s_{23}s_5s_6\right] + c_1\left(s_4c_5s_6 + c_4c_6\right), \\
o_z &= -s_{23}\left(c_4c_5s_6 + s_4c_6\right) + c_{23}s_5s_6, \\
a_x &= -c_1\left(c_{23}c_4s_5 - s_{23}c_5\right) - s_1s_4s_5, \\
a_y &= -s_1\left(c_{23}c_4s_5 - s_{23}c_5\right) + c_1s_4s_5, \\
a_z &= -s_{23}c_4s_5 - c_{23}c_5.
\end{aligned} \tag{4}$$

Where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{ij} = \sin(\theta_i + \theta_j)$, $c_{ij} = \cos(\theta_i + \theta_j)$.

According to Eq. (1) and (4), because all the 6 joints of the robot are rotary joints, the variables of D-H parameters are $\theta_i (i = 1 \sim 6)$. If all the θ_i are known, it is easy get T with Eq. (4). When a 6 rotary joints robot is assembled, a_i 、 α_i and d_i are fixed, and we assume they are accurate, while θ_i are variables which can be determined by angle encoder. But angle encoder always has zero error, θ_i in Eq. (1) and (4) must be replaced by θ_i , where $\theta_i^{'} = \theta_i + \Delta\theta_i$, and $\Delta\theta_i$ is zero error modifier. In order to improve positioning accuracy, $\Delta\theta_i$ must be found. This paper will introduce how to get $\Delta\theta_i$ based on improved FA.

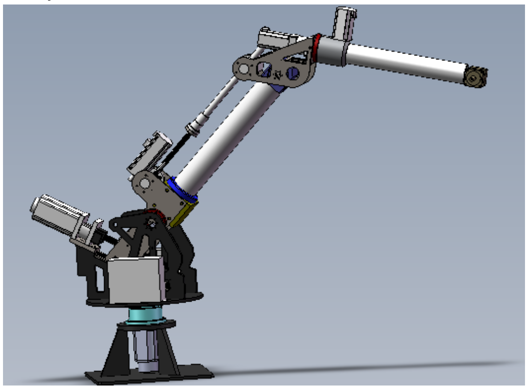

Fig. (1). 6 joints robot 3D simulation diagram

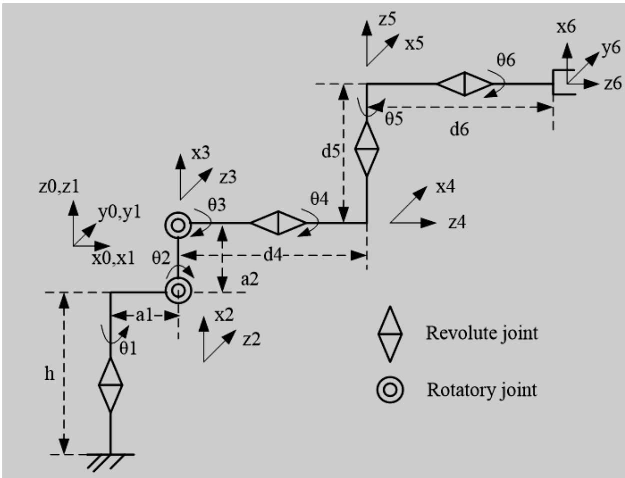

Fig. (2). 6 joints robot schematic diagram

III BASIC FIREFLY ALGORITHM

Firefly algorithm is a heuristic algorithm which is developed in 2008 by Xin-She Yang. The algorithm mimics the firefly's behavior to find food by emitting light. Some characteristics of the firefly are abandoned. For example, the sex of the firefly is ignored. The firefly uses light emission features to search for a nearby area and moves to the position which is better. In the optimizing process, the brightness and attraction of firefly are two key factors for the firefly and position evolution. The position of the firefly depends on its brightness. The higher brightness, the better position. The attraction of the firefly determines the moving direction of the firefly. The more attractive the firefly is, the easier it is to attract the firefly to its own position. The brightness and attraction of the firefly are inversely proportional to the distance between two fireflies, and they decrease with the increase of the distance. The effect of the algorithm is achieved by updating the attraction and brightness constantly. The mathematical description of the firefly algorithm is as follows. [3, 4, 11]

Supposed there are m fireflies and the dimension of search space is N, the initial position of the firefly i is $x_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$.

Definition 1. The relative brightness of the firefly i is as follow:

$$I_i = I_{0i} \cdot e^{-\gamma \cdot r_{ij}}. \quad (5)$$

Where γ is the attraction factor of the light intensity,

(4) Update and save the best firefly position and brightness;

(5) If the stopping criterion is met, then the best global position and brightness are found, otherwise return to step (2).

IV. IMPROVEMENT OF BASIC FIREFLY ALGORITHM

Basic FA is easy to realize, but it exists premature

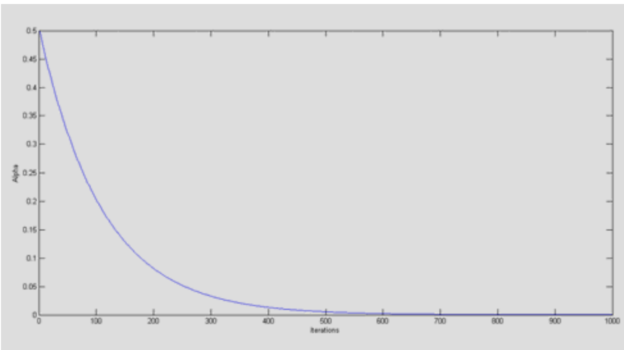

Fig. (3). α curve

which value is related to the scale area and set constant in generally. r_{ij} is the distance between firefly i and j, where $r_{ij} = |x_i - x_j|$, The value I_i decreases with the increasing of r_{ij}, when $r_{ij} = 0$, I_i will reach the maximum value I_{0i} which value is generally proportional to the objective function.

Definition 2. The attractiveness is as follow:

$$\beta = \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2}. \quad (6)$$

where β_0 is the attractiveness at r = 0.

Definition 3. The movement of firefly i attracted to

$$x_i(t+1) = x_i(t) + \beta \cdot (x_j(t) - x_i(t)) + \alpha \cdot (rand - 0.5). \quad (7)$$

Where, α is the step size, rand is a random factor that is subject to the uniform distribution [0, 1]. In order to increase the spatial range of the search space, improve the diversity of the population and prevent premature convergence, a random disturbance $\alpha \cdot (rand - 0.5)$ is brought in.

The step of basic firefly algorithm is as follows: [3, 4, 11, 12]

(1) Initialize the population of the fireflies randomly;

(2) Calculate the brightness of each firefly;

(3) Take comparison of the brightness. If firefly j is brighter than firefly i, firefly i will move to the area of firefly j based on Eq. (7).

(4) Update and save the best firefly position and brightness;

(5) If the stopping criterion is met, then the best global position and brightness are found, otherwise return to step (2).

Basic FA is easy to realize, but it exists premature phenomenon. For multi-dimensional and multi-peak procedure, the basic algorithm easily converges to the local optimum. In order to overcome the disadvantage, a lot of literatures are proposed to improve the algorithm [11-14]. This paper will use several improvements below:

1) Set the exponential declining step factor α

The term $\alpha \cdot (rand - 0.5)$ is in order to increase the diversity of the population. When α is large, it is advantageous to increase the sample diversity, expand the search scope, and avoid quickly to enter the local optimal value. When the α is small, it is good for fine search. In the algorithm, α should be diminishing. The iteration of α is:

$$\alpha = \alpha \cdot \left(10^{-4} / 0.9\right)^{1/MaxIteration}. \quad (8)$$

Where, MaxIteration is the maximum iterations.

When $\alpha_0 = 0.5$, the curve of Eq. (8) is as Fig.(3).

2) Optimize position update equation

In order to improve the convergence of the algorithm, the

$$x_i(t+1) = x_i(t) + \omega \cdot rand \cdot (x_{best}(t) - x_i(t)) + \beta \cdot (x_j(t) - x_i(t)) + \alpha \cdot (rand - 0.5). \quad (9)$$

Comparison with Eq. (7), Eq. (9) has an additional term $\omega \cdot rand \cdot (x_{best}(t) - x_i(t))$, where $x_{best}(t)$ is the position of the firefly whose brightness is the best; rand is a random factor that is subject to the uniform distribution [0, 1]; ω is a weight factor, ω has a similar recursive equation with α shown as Eq. (10), where $\omega_0 = 0.9$.

$$\omega = \omega \cdot \left(10^{-4} / 0.9\right)^{1/MaxIteration}. \quad (10)$$

3) Bring in simulated annealing mechanism

Simulated annealing algorithm (SA) is a random combination optimization method which developed in the early 1980s. It simulates the thermodynamic process of metal temperature cooling, and is widely used in combination optimization problem. The initial temperature is determined by the simulated annealing algorithm in the optimization, randomly select an initial state and investigate the objective function of the state; Attach a small perturbation to the current state and calculate the target function values for the

DOI 10.5013/IJSSST.a.17.02.08 ISSN: 1473-804x online, 1473-8031 print

new state; In the whole cooling process, the better state will be accepted in probability 1, while worse state will also be accepted in some probability which is less than 1. If the start temperature is high enough and the temperature falls slowly enough, SA can converge to the global optimum with probability 1. Because of its ability to accept the worse sample in some probability and improve the sample's diversity, it has the ability to jump out of the local optimal solution. [15]

In Eq. (8), in order to improve the diversity of the population, $x_{best}(t)$ is selected in the follow FA.

$$\left(\min(\mathbf{I}) - I_{best}\right) < \varepsilon \,\|\, e^{-\left(\frac{\min(\mathbf{I}) - I_{best}}{T}\right)} > rand. \tag{11}$$

Where, the brightness and the target function are positive ratio, the smaller the target function value indicates the higher the brightness. ε is a decimal in [0, 1] which is to improve the diversity of the population. rand is a random factor that is subject to the uniform distribution [0, 1]. T is the annealing temperature.

V. FIND ROBOT ZERO ANGLE ERRORS WITH IMPROVED FA

The steps to find robot zero angle errors $\Delta\theta_i$ with improved FA are as follows:

(1) Drive the robot to any N positions with the controller, then get these N positions' coordinates with some equipment, read and record the corresponding angles θ_{ji} in controller, where j=1~N, i=1~6

(2) Consider the 6 expected zero angle errors as the fireflies' position. At the begin, initialize all the fireflies' positions, in other words, each firefly has 6 random zero angle errors $\Delta\theta_{mi}$, where m is the number of the fireflies, i=1~6.

(3) Calculate theoretical values of these N positions with θ_{ji} in step (1) and $\Delta\theta_{mi}$ in step (2).

(4) Calculate the differences of actual values and theoretical values for each position, as shown in Eq. (12) or Eq. (13).

$$F_i = \sum_{m=1}^{3} \left| T_{m4} - T_{im4} \right|. \tag{12}$$

$$F_i = \sum_{m=1}^{3}\sum_{n=1}^{4} \left| T_{mn} - T_{imn} \right|. \tag{13}$$

Where T is the actual pose matrix, T_{mn} is the element of T; T_i is the calculation pose matrix, T_{imn} is the element of T_i.

The difference between Eq. (12) and (13) is that Eq. (12) is concerned only at the distance of the robot terminal position, while Eq. (13) pays attention to the distance of the terminal position and direction at the same time.

(5) Calculate the average value $\bar{F}_i$ of F_i, and consider $\bar{F}_i$ as the firefly's brightness. Save the current best brightness and position.

(6) Update the positions with Eq. (11).

(7) If the brightness of firefly j is better than brightness of firefly i, then firefly i move to the region near firefly j in a manner as Eq. (9).

(8) Calculate brightness again, and save the current best brightness and position.

(9) If the stopping criterion is met, then the best global brightness and position are found, otherwise, return to step (3).

VI. SIMULATION AND VERIFICATION

In simulation, 6 small random zero angle errors $\Delta\theta_i$ are generated from a uniform distribution [-0.1, 0.1], here they are set to [-0.1336 -0.1856 +0.0116 +0.1728 -0.0688 -0.1301](Unit: º) simply for ease of comparison. N teams of joint thetas θ_{ji} are generated from a uniform distribution [-180, 180] (Unit: º). Then the pose matrices are calculated with Eq. (4) based on $\Delta\theta_i$ and θ_{ji}, which are considered as the robot pose matrices.

In the improved FA, the population size is 40, the dimension is 6, iterations is 1000, $a_0 = 0.5$, $\omega_0 = 0.9$, $\varepsilon = 0.01$, annealing initial temperature $T_0 = 1000$, temperature cooling rate $K = 0.99$. The simulation results with different position numbers N are shown in table 1 and 2, where $\Delta\theta_i (i = 1 \sim 6)$ are zero angle errors searched in improved FA, F is the averages of distances between each pair of theoretical positions and actual positions before calibration, while F' is the averages after calibration. F'/F is the multiple of the position distance errors before and after calibration.

TABLE 1. IMPROVED FA SIMULATION RESULT (WITH EQ. 12)

$\Delta\theta_i (i = 1 \sim 6)$	N	F	F'	F' / F
0.0983 -0.2008 0.5355 -0.9320 -0.3880 0.3670	1	0.6153	2.9781e-04	4.84E-04
0.1336 -0.0856 0.1212 0.0727 -0.1029 0.1411	2	0.4977	5.2745e-05	1.06E-04
0.1336 -0.0856 0.1211 0.0730 -0.1029 0.5992	3	0.7412	1.0101e-04	1.36E-04
0.1336 -0.0856 0.1212 0.0728 -0.1027 0.4432	4	0.6990	6.1718e-05	8.83E-05
0.1336 -0.0856 0.1212 0.0728 -0.1028 0.0608	5	0.6112	2.0079e-05	3.29E-05
0.1336 -0.0856 0.1212 0.0729 -0.1027 -0.3990	6	0.7909	7.2641e-05	9.18E-05
0.1336 -0.0856 0.1211 0.0729 -0.1027 -0.7187	7	0.8731	8.6155e-05	9.87E-05
0.1336 -0.0856 0.1212 0.0727 -0.1028 0.8683	8	0.6421	3.8888e-05	6.06E-05
0.1336 -0.0856 0.1212 0.0728 -0.1027 -0.7147	9	0.8392	8.1438e-05	9.70E-05
0.1336 -0.0856 0.1212 0.0727 -0.1028 0.0031	10	0.7021	5.9383e-05	8.46E-05

TABLE 2. IMPROVED FA SIMULATION RESULT (WITH EQ. 13)

$\Delta\theta_i (i = 1 \sim 6)$	N	F	F'	F' / F
0.0965 -0.0159 0.1689 -0.3851 -0.1830 -0.6641	1	0.7223	0.0246	0.0341
0.1341 -0.0851 0.1200 0.0775 -0.1030 -0.2038	2	0.4619	0.0044	0.0095
0.1332 -0.0856 0.1188 0.0897 -0.1033 -0.1878	3	0.8749	0.0057	0.0065
0.1336 -0.0856 0.1213 0.0729 -0.1029 -0.2408	4	1.0845	0.0060	0.0055
0.1336 -0.0856 0.1212 0.0727 -0.1027 -0.1005	5	0.8136	0.00196	0.0024
0.1336 -0.0856 0.1212 0.0727 -0.1028 -0.1142	6	0.8366	5.3742e-04	0.0006
0.1336 -0.0856 0.1212 0.0731 -0.1028 0.0949	7	0.9377	0.0114	0.0122

According to table 1, when N changes from 1 to 10, all the F' / F greatly reduces. When N changes from 1 to 5, F' / F is decreased. When N changes from 6 to 10, F' / F has fluctuated. The value of F' / F is the smallest when N=5. From the start of N=2, the search value of the first 5 zero errors are close to the actual values, while the 6th is random. The reason why the 6th is random is that the 6th joint is a revolute joint, its revolute angle θ_6 just affects its direction but not affects its position, and Eq. (12) can only represents the error of position. In the algorithm with Eq. (12), the optimal value of the 6th joint zero angle error can't be searched. There are just 5 variables in Eq. (12). It can also explain why N=5 can reach the best result.

The characteristics of the table 2 are basically similar to table 1. The only difference is that all the 6 zero angle errors can be accurately searched in table 2 when N=6 because Eq. (13) can represent the error of position and direction at the same time.

Eq. (12) and Eq. (13) each has its advantages and disadvantages. When using Eq. (12), we just need a three coordinate tester to get the robot terminal position coordinates, and it is simple and cheap, but the 6th zero angle error can't be searched. When using Eq. (13), we need a double theodolite tester to get the robot terminal position coordinates and directions, and it is complex and expensive, but all the zero angle errors can be searched. In order to get a better result, when using Eq. (12) set N=5, set N=6 while using Eq. (13).

In this algorithm, the brightness curve throughout the iteration is shown in Fig. (4). According to the figure, the search value is close to the optimal value at the iteration 550. In the first 550 iterations, search values quickly close to the optimal values, and then converges slowly to the optimal value after 550 times.

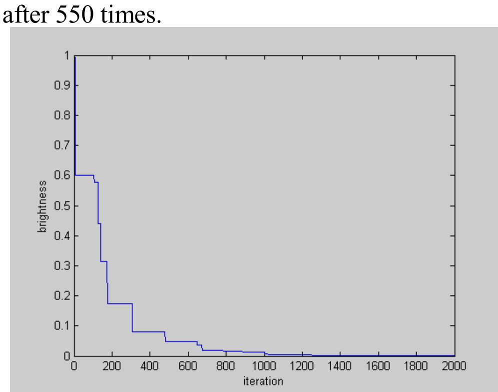

Fig. (4). Brightness curve

VII. CONCLUSION

The absolute accuracy of the robot is affected by many aspects. The joints' zero angle errors are one of them. With the improved FA introduced in this paper, the optimal values of the zero angle errors can be found quickly. In this algorithm, just a few position messages need to be obtained. It is easy to operate. This method has good generality. It not only can be applied to the robot calibration all with revolute joints, but also can be applied to the robot calibration with sliding joints.

REFERENCES

[1] Wang DongShu, Chi JianNan. "Survey on robot kinematics calibration", Application Research of Computers, vol. 24(9), 2007, pp. 8-11.

[2] Wang RuiFang, Xu Fang. "Design and Research on Motion Controller of Grinding and Polishing Robot". Microcomputer Information, vol. 23(9-2), 2005, pp. 261-263.

[3] M.Grotjahn, M.Daemi, D.Heimann. "Friction and rigid body identification of robot dynamics", International journal of solids and structures", vol. 38, 2001, pp. 1889-1902.

[4] Veitschegger W K, Wu C H. "Robot calibration and compensation", Robotics and Automation, IEEE Journal, vol. 4(6), 1988, pp. 643-656.

[5] Renders J.M, E Rossignal, M.Becquet, R.Hanus. "Kinematic calibration and geometrical parameter identification for robots", IEEE transactions on robotics and automation, vol. 7, 1991, pp. 721-731.

[6] Hanqi Zhuang, Kuanchih Wang. "Optimal planning of robot calibration experiments by genetic algorithms", Proceedings of the 1996 IEEE international conference on robotics and automation, Minneapolis, Minnesota, 1996, pp. 981-986.

[7] Hanqi Zhuang, Kuanchih Wang. "Optimal selection of measurement configurations for robot calibration using simulated annealing", IEEE transactions on robotics and automation, 1994, pp. 393-398.

[8] Yang Xin-She. "Nature-Inspired Metaheuristic Algorithm", Luniver Press, 2008.

[9] Yang Xin-She. "Cuckoo Search and Firefly Algorithm Theory and Applications", Springer, 2014.

[10] Han JianHai. "Industrial robot", Huazhong University of Science and Technology press, 2009.

[11] Wang MingBo, etc.. "Multi-group firefly algorithm based on simulated annealing mechanism", Journal of Computer Applications, vol. 35(3), 2015, pp. 691~695.

[12] Wang Xiang, etc.. "A novel improved firefly algorithm", Computer and Applied Chemistry, vol. 31(8), 2014, pp. 987~992.

[13] Srivatsava P R, Mallikarjun B, Yang Xin-she. "Optimal test sequence generation using firefly algorithm Original Research Article", Swarm and Evolutionary Computation, vol. 8(1), 2013, pp. 44~53.

[14] Kazem A, Sharifi E, Hussain F K, et al. "Support vector regression with chaos-based firefly algorithm for stock market price forecasting", Applied Soft Computing, vol. 13(2), 2013, pp. 947-958.

[15] Gao Ying, Xie ShengLi. "Particle Swarm Optimization Algorithms Based on Simulated Annealing", Computer Engineering and Application, vol. 4(1), 2004, pp. 47~50.

DOI 10.5013/IJSSST.a.17.02.08 ISSN: 1473-804x online, 1473-8031 print