Inverse Kinematics of Multi-joint Robot Based on Particle Swarm Optimization Algorithm

Ze Fan Cai [1,2], Dao Ping Huang [2], and Yi Qi Liu [2]
[1] Electronics and Information Engineering Department of Shunde Polytechnic, Foshan, China
[2] Automation College of South China University of Technology, Guangzhou, China
Email: 12266423@qq.com, audhuang@scut.edu.cn, 653830373@qq.com

***Abstract*—The optimization of the inverse kinematics of robot is the most important part of robot kinematics. Parameters for each link of the robot are needed to be identified according to the position and orientation in both of teaching programming and interpolation. In view of the fact that to get the analytic solutions of the inverse kinematics of robots must meet special conditions, this paper used a general solution, simulated annealing particle swarm optimization algorithm, to obtain the optimization of inverse kinematics without any special conditions. Simulation results show that the algorithm can be effectively applied to the robot inverse kinematics and is able to achieve sufficient accuracy.**

***Index Terms*—particle swarm optimization algorithm, simulated annealing algorithm, multi-joint robot, inverse kinematics, optimization**

I. Introduction

Kinematics of multi-joint robot includes forward kinematics and inverse kinematics. The forward kinematics is to obtain the robot pose based on the robot joint variables. Instead, the inverse kinematics is to obtain the robot joint variables based on the robot pose. [1] The solution of forward kinematics can be obtained by the pose matrix, and the solution is analytical, determined, and unique. The best method for inverse kinematics is to get the analytical solution of joint variables according to the inverse kinematics equation, but the analytical solution can only be found out in the certain conditions [2], otherwise, the optimal solution can just be obtained by various optimization algorithms.

Since 1980s, many bionic optimization algorithms have developed rapidly, such as artificial neural network (ANN), genetic algorithm (GA), ant colony (ACO), simulated annealing (SA) and tabu search algorithm (TS). They were all developed by simulating or revealing certain phenomena or processes of nature. They are also called intelligent optimization algorithms.

Particle swarm optimization algorithm (PSO) [3] is one of the bionic optimization algorithms. PSO is a swarm intelligence algorithm that mimics the foraging behavior of birds which was developed by two American scholars Kennedy R. and C. Eberhart J in 1995. Because PSO is simple and easy to be realized, it has caused the attention of scholars, and hundreds of papers about PSO have been published. It has been applied successfully in many fields. [4] In this paper, simulated annealing mechanism is introduced to improve the standard PSO. This paper tries to obtain the optimization of inverse kinematics with this improved PSO. Simulation result shows that the algorithm can be effectively applied to the robot inverse kinematics and is able to achieve sufficient accuracy.

II. Denavit-Hartenberg Parameters Expression of Multi-joint Robot

There are several robot pose expressions, such as Euler angle, rotation matrix, four element method, etc. The most common one is the rotation matrix, and Denavit-Hartenberg (D-H) parameters expression is the most famous one of the rotation matrix methods. D-H was developed by Denavita and Hartenberg in 1995. [5]

The D-H parameters of the link i must be determined firstly. The 4 parameters are link length a_i, link twist α_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 in

$$T_i = \begin{bmatrix} c_i & -s_i\cos\alpha_i & s_i\sin\alpha_i & a_i c_i \\ s_i & c_i\cos\alpha_i & -c_i\sin\alpha_i & a_i s_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$.

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 6^{th} joint) and base coordinate system. T is as in [5]

Manuscript received June 25, 2015; revised October 21, 2015.

doi: 10.18178/joace.4.4.306-308

$$T = T_1 T_2 T_3 T_4 T_5 T_6. \quad (2)$$

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 in

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

Fig. 1 is the three-dimensional (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 in (3). ***p***、***n***、***o*** and ***a*** in this equation are as in (4).

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

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 (1) and (4), because all the 6 joints of the robot are rotary joint, the variables of D-H parameters are $\theta_i (i = 1 \sim 6)$. If all the θ_i are known, it is easy to get T with (4). According to [2], the solution of the inverse kinematics of the 6 DOF robot can be obtained only when at least one of the two conditions in the Pieper criterions is met. The two Pieper criterions are:

(A) Three adjacent joint axes cross the same point;

(B) Three adjacent joint axes are parallel to each other.

The robot described in this paper can't meet the above two criterions, so the optimal solution can be obtained only by numerical method.

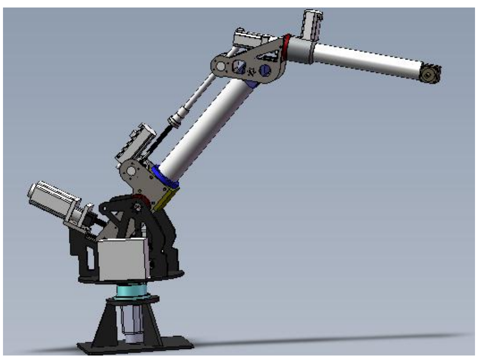

Figure 1. 6 joints robot 3D simulation diagram

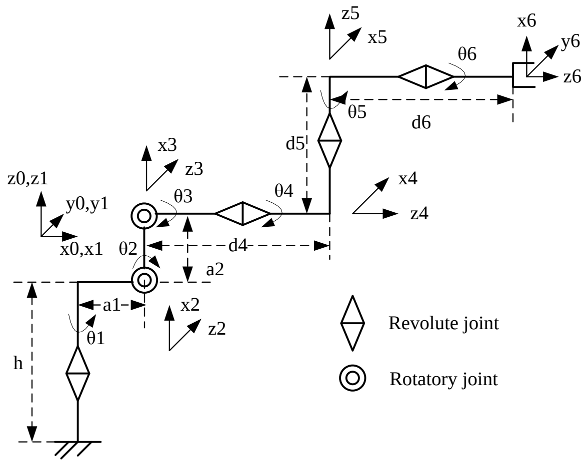

Figure 2. 6 joints robot schematic diagram

III. Basic Particle Swarm Optimization Algorithm

PSO is one of the bionic optimization algorithms. PSO is a swarm intelligence algorithm that mimics the foraging behavior of birds which was developed by two American scholars Kennedy R. and C. Eberhart J in 1995. The particle swarm optimization algorithm is realized by iteration. In each iteration the particles update their speeds and positions by tracking their own optimal value pBest and global optimal value gBest. The speed update equation is as in (5) and the position update equation is as in (6).

$$\begin{aligned} v(t+1) &= W \cdot v(t) + c_1 \cdot \alpha \cdot \left[pBest(t) - x(t)\right] \\ &\quad + c_2 \cdot \beta \cdot \left[gBest(t) - x(t)\right] \end{aligned} \quad (5)$$

$$x(t+1) = x(t) + \gamma \cdot v(t+1) \quad (6)$$

where W is the coefficient of keeping the original speed, called inertia weight.

c_1 is the weight coefficient of the optimal value of the particle itself, which represents the particle's own cognition, and is usually set to 2.

c_2 is the weight coefficient of the global optimal value of the whole population, which represents the knowledge of the entire group of particles, and is also usually set to 2.

α and β are random factors that are subject to the uniform distribution [0, 1].

γ is a constraint factor, and usually set to 1.

IV. Obtain the Solution of the Multi-joint Robot Inverse Kinematics Based on SA-PSO

Although the concept of PSO algorithm is simple and needs fewer parameters to adjust, and easy to be realized, it is easy to converge to local optima, so PSO is often used together with other optimal algorithms in order to improve the ability to converge to the global optimum.

Simulated annealing algorithm 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 value for the 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. [6]

The researchers combine the strong global convergence of SA with the simple and efficient performance of PSO, and get the simulated annealing particle swarm optimization (SA-PSO), and successfully apply it to many subjects. [6-8]

In this paper, SA-PSO is applied to the solution of the inverse kinematic of the 6-joint robot. The result is satisfying.

The solution T of forward kinematic is obtained with (1) ~ (4). The fitness function of SA-PSO is as in

$$p_i = \sum_{m=1}^{3}\sum_{n=1}^{4}\left|T_{mn} - T_{imn}\right| \tag{7}$$

where T is the target pose matrix which is the expected pose matrix of the robot in a certain time, T_i is the actual pose matrix of the certain particle, p_i is the deviation of the corresponding elements of the actual and expected pose matrix. Since the value of the last row of the matrix T_i in (2) is kept constant, last row is not within the range of computation.

The steps of SA-PSO are as follows [9, 10]:

(A) Randomly initialize the position and speed of each particle, calculate the fitness of each particle and regard the fitness as the local particle optimal fitness p_i. Take the smallest p_i as the global optimal fitness p_g.

(B) Initialize the start temperature T and cooling rate K.

(C) Obtain an alternative for the global optimum. The optimal particle that satisfies the follow equation is used as an alternative for the global optimum.

$$\left(\min\left(p\right) - p_{best}\right) < \varepsilon \,||\, e^{-\left(\frac{\min(p) - p_{best}}{T}\right)} > rand. \tag{8}$$

where 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.

(D) Update the speeds and positions with (5) and (6), and make sure that the speeds and positions are not exceeded the maximum.

(E) Calculate the new fitness $pTemp_i$ of each particle, and consider the $pTemp_i$ meeting (9) as the new p_i.

$$\left(pTemp_i - p_i\right) < \varepsilon \,||\, e^{-\left(\frac{pTemp_i - p_i}{T}\right)} > rand \tag{9}$$

where, ε and $rand$ are the same as in (8).

(F) Update p_g.

(G) Take simulated annealing operation.

(H) If the stopping criterion is met, then the best global fitness and position are found, otherwise, return to step (C).

From the above steps, we can see that simulated annealing operation is taken in step (C) and (E). In other words, SA-PSO takes simulated annealing operation twice in every iteration and enhance the algorithm converges to the global optimal value.

V. Simulation and Verification

To verify the feasibility of the algorithm, the inverse kinematic matlab simulation for the 6 DOF robot shown in Fig. 1 is given. In order to make ***p***, ***n***, ***o*** and ***a*** in the pose matrix T the same magnitude, take the unit of each link parameter as meter. In the simulation, $a_1 = 0.232$, $a_2 = 1.000$, $d_4 = 0.600$, $d_5 = 0.043$, $d_6 = 0.025$. The random expected pose matrix is as follow:

$$T = \begin{bmatrix} -0.612372 & 0.612372 & -0.500000 & 1.359193 \\ 0.707106 & 0.707106 & 0.000000 & 0.030405 \\ -0.353553 & 0.353553 & 0.866025 & -0.013167 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

In this SA-PSO, the population size is 40, the dimension is 6, $c_1 = c_2 = 2$, $\varepsilon = 0.01$, W is as in (11).

$$W = MaxW - CurLoop \times (MaxW - MinW) / TotLoop \tag{11}$$

where, CurLoop is the current iteration number, TotLoop is the maximum iteration number which is set to 500, annealing start temperature $T_0 = 100$, cooling rate $K = 0.99$.

After 500 iterations, the pose matrix is as follow:

$$T' = \begin{bmatrix} -0.612372 & 0.612372 & -0.500000 & 1.359183 \\ 0.707106 & 0.707106 & 0.000000 & 0.030405 \\ -0.353553 & 0.353553 & 0.866025 & -0.013177 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

In the simulation, the minimum of p_i is 0.000020. The fitness curve is shown as in Fig. 3.

According to (10) and (12), the direction parts of T and T' completely anastomose, while two components of the position parts have errors 0.000010m (0.010mm). The errors meet the requirements of robot pose accuracy. The simulation result for all the 500 iterations is shown in Fig. 3. From Fig. 3, it can be seen that the fitness is quickly close to the global optimum in exponential pattern in the first 15 iterations, and continue to move to the global optimum very slowly after 15 iterations; There are lots of peaks and troughs in the curve of surrogate value of the global optimal value. Each trough means a local optimal value. The simulation result shows that SA-PSO can effectively jump out of the local optimal value and has the strong ability to achieve the global optimal value.

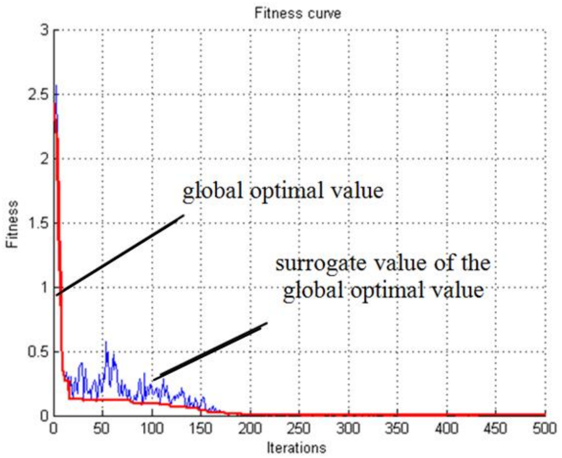

Figure 3. Iteration curve

VI. Conclusion

The forward kinematics and inverse kinematics are two aspects of robot kinematics. Based on D-H, the forward kinematics equation and pose matrix can be obtained quickly. But the solution of the inverse kinematics is not easy to obtain. If one or two conditions in the Pieper criterions is met, the analytical solution can be obtained by using the analytical method, otherwise the optimal solution can only be obtained by numerical solution. Because the solution of robot inverse kinematics has a lot of local optimal solutions, the numerical solution is required to have a strong ability to obtain the global optimal value. After working with simulated annealing mechanism, PSO algorithm overcomes the disadvantages of being easily trapped into a local optimal solution, it can effectively reach the global optimal value. If the pose matrix is known, the SA-PSO algorithm can search the angle values of every joint, and thus the expected position and direction can be obtained accurately enough. The simulation result shows that the algorithm can be applied to inverse kinematics of multi DOF robot.

References

[1] X. C. Xie and T. Zhang, *Robotics and Its Application*, Press of Machinery Industry, 2012, ch. 2, pp. 44-54.
[2] Y. L. Xiong, *Robot Technology Foundation*, Huazhong University of Science and Technology Press, 1996, ch. 3, pp. 46.
[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at the IEEE International Conference on Neral Networks, Perth Western Australia, 1995.
[4] L. Li and B. Niu, *Particle Swarm Optimization Algorithm*, Metallurgical Industry Press, 2009, ch. 2, pp. 22-32.
[5] J. H. Han, *Industrial Robot*, Huazhong University of Science and Technology Press, 2009, ch. 3, pp. 50-74.
[6] Y. Gao and S. L. Xie, "Particle swarm optimization algorithms based on simulated annealing," *Computer Engineering and Application*, vol. 4, no. 1, pp. 47-50, 2004.
[7] Y. Li, J. J. Wang, L. H. Cao, "Simulated annealing particle swarm optimization algorithm of optimal load dispatch in power plan," in *Proc. the CSU-EPSA*, vol. 23, no. 3, pp. 40-44, 2011.
[8] Y. Zheng, W. M. Cheng, Y. Cheng, and X. Wu, "Reliability optimization for cylindrical gear reducer based on simulate Anneal particle swarm optimization," *Mechanical Drive*, vol. 34, no. 10, pp. 43-47, 2010.
[9] Y. Sun, X. K. Wang, D. B. Guo, and Y. H. Zhang, "The optimized algorithm for data reconciliation based on particle swarm optimization," *Journal of Test and Mearsurement Technology*, vol. 29, no. 2, pp. 149-157, 2015.
[10] W. Li, L. Zhou, and X. L. Chai, "Joint application of SA-PSO and PDA in sensor registration," *J. Zhengzhou Univ.*, Nat. Sci. ed., vol. 47, no. 1, pp. 69-73, 2015.

Ze Fan Cai was born in Guangdong, China in 1979 and received B.S in Automation in 2003 and M.S degree in Control Theory and Control Engineering in 2006 from South China University of Technology, Guangzhou, China. He now is a lecturer in Shunde Polytechnic, Foshan, China and also a PhD's Student of Detection Technology and Automation Equipment at South China University of Technology, Guangzhou, China. His research focuses on wireless network control, robot control and embedded control.

Dao Ping Huang received his B.S degree in 1982, M.S degree in 1986 and PhD degree in 1998 from South China University of Technology, Guangzhou, China. He obtained his Professor title in 2001 of South China University of Technology. He has published more than 70 papers in academic journals. His research focuses on intelligent detection and control, soft sensor technology, fault diagnosis and accident forecast for industrial process.

Yi Qi Liu received his B.S. and M. S. degree from Beijing University of Chemical Technology in 2006 and 2009, P. R. China, respectively. His PhD obtained from South China University of Technology in 2013, P. R. China. During this period, he joined AWMC, University of Queensland, Australia, as a joint training PhD supporting by Chinese Scholarship Council (CSC) from 2011.10-2013.04. Up to date, he served as a lecturer at South China University of Technology. During 2015. 08-2015.11, he visited TU Ilmenau, Germany, supported by DAAD (Deutscher Akademischer Austausch Dienst). He has published more than 25 papers in the academic journals. His research focuses on soft-sensors, fault diagnosis and optimized control.