Zhang *et al. Robot. Biomim.* (2018) 5:3
https://doi.org/10.1186/s40638-018-0085-7

Robotics and Biomimetics

RESEARCH
Open Access

A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm

Jiayan Zhang, Qingxi Meng*, Xugang Feng and Hao Shen

Abstract

By using quintic polynomial function to interpolate several given points of each joint of the robot, the mathematical expressions of each joint variable of the robot with time are established. In addition, to improve the search algorithm performance crossover operator and mutation operator of the genetic algorithm are improved in cosine form. Furthermore, the improved adaptive genetic algorithm is applied to optimize the time interval of interpolation points of each joint, so as to realize time optimal trajectory planning. Moreover, MATLAB simulation is carried out, and the results show that the method proposed in this paper reduces the running time of the robot tasks. Meanwhile, the curves of angle position, velocity and acceleration of each joint are smooth enough, which ensure accomplish its tasks in a stable and efficient way.

Keywords: Industrial robot, Trajectory planning, Adaptive genetic algorithm (AGA), Time optimal

Background

Robot trajectory planning usually refers to track points given several expectations and target pose, and timely adjust the rotation angle of each joint of the robot to the end effector at a prescribed trajectory followed by each point to eventually reach the target point. The trajectory planning in joint space is simpler and convenient than that of Cartesian space. Therefore, several fixed points which located at the end of several robotic arms are usually given. Then, these track points for the robot are computed by using the inverse kinematics so as to convert it from Cartesian space to joint coordinate space. Next, the track points are used to carry out interpolation operation using various spline functions, polynomial functions or other forms of curves, and the expressions about the time of each joint variable for the robot are obtained. In addition, in light of the mechanical characteristics of the robot, the speed and acceleration of each joint should be limited to the allowable range. Therefore, it is necessary to optimize the speed and acceleration of each joint arm, not only to ensure the smooth operation of the joint arm but also to reduce the wear and impact to prolong the working life of the robot.

The method of optimal trajectory planning generally includes time optimal trajectory planning [1–3], energy minimum trajectory planning [3, 4] and impact minimum trajectory planning [5], or multi-objective trajectory optimization combining these estimation schemes. Among them, the optimal trajectory planning with the robot running time as main consideration is favored by many scholars. In recent years, many researchers have made some achievements in the field of robot trajectory planning. Tohfeh and Fakharian [6] constructed a function expression for the omnidirectional robot's energy dissipation by combining obstacle avoidance performance, and the optimization problem was transformed into a parameter minimization problem by the polynomial interpolation method, which provided a more effective way for the study of robot obstacle avoidance. However, due to the complexity of this method, there is a certain degree of difficulty in practice. Bende [7] studied a method of modeling underwater robot based on bond graph theory and optimized the model parameters with the genetic algorithm to obtain the optimized trajectory

*Correspondence: mqx3021@qq.com
College of Electrical Engineering and Information, Anhui University of Technology, Ma'anshan 243002, China

of the robot. Experiments were carried to show the significance of this method. Zhu and Liu [8] applied the seventh-order B-spline curve for the interpolation operation of robot's articulation arm trajectory and applied the sequential quadratic programming for the trajectory planning, which achieved the optimal planning goal and the angular displacement, velocity and acceleration curve of each joint of the robot are smoother.

In this paper, from the point of view of robot running time, the trajectory of robot joint is planned by the quintic polynomial interpolation under the premise of smooth operation of the manipulator, and the time interval of trajectory interpolation point is optimized by the improved adaptive genetic algorithm, so that the robot accomplishes task time as short as possible.

Model

In this paper, we consider PUMA560 robot as the research object, which belongs to a conventional six-arm-type robot. Its first three joints are used to determine the position of the robot end effector, and the latter three joints are used to determine its attitude. In order to make it easier to analyze the motion of articulation arm of the robot, it is necessary to establish the kinematic equation of the robot. The link pole coordinate system is established by D-H method, and through the translation, rotation and other coordinate transformation to establish the relationship between the coordinate system. The link pole coordinate system is shown in Fig. 1.

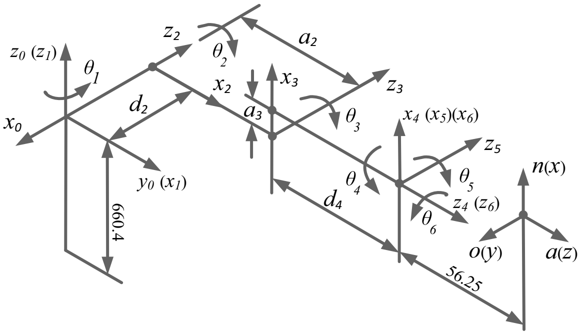

Fig. 1 The link pole coordinate system of PUMA560 robot according to the link pole coordinate system of Fig. 1, the transformation matrix of each link can be obtained as follows

$$
{}^0T_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1T_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s\theta_2 & -c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^2T_3 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3T_4 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^4T_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5T_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

Multiplying the above transformation matrice in turn, the comprehensive transformation matrix of the PUMA560 robot is obtained:

$$
{}^0T_6 = {}^0T_1(\theta_1){}^1T_2(\theta_2){}^2T_3(\theta_3){}^3T_4(\theta_4){}^4T_5(\theta_5){}^5T_6(\theta_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

where the third-order sub-matrix $\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$ represents the posture of the end effector relative to the base system {0}, and $\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$ represents its position relative to the base system {0}.

Trajectory planning analysis

The number of desired track points of the robot end effector in the Cartesian space is transformed into the corresponding joint variables by the kinematic inverse operation, and then, track points of the joint space are interpolated to obtain the joint variables of the robot a function of time-varying expression [9]. In this paper, the quintic polynomial is used to interpolate trajectory points in the robot joint space.

Denote by $\theta(t)$ the joint angle. Assume $\theta(t_0){=}\theta_0$, $\theta(t_f){=}\theta_f$. Obviously, there are several quintic polynomial curves satisfying the above conditions, as shown in Fig. 2.

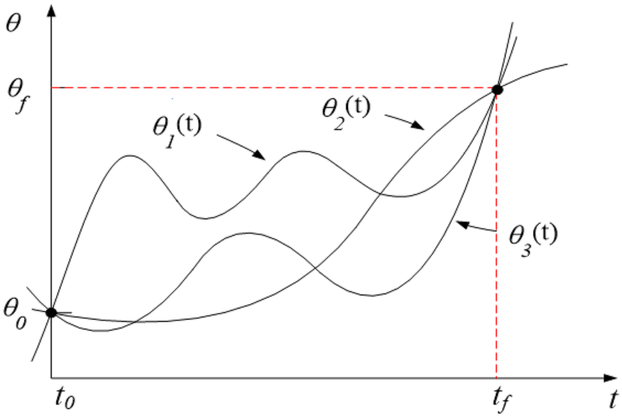

Fig. 2 Quintic polynomial function of the same starting point and end point

It is now necessary to find a smooth curve with starting point θ_0 and ending point θ_f. In addition, the expression of the quintic polynomial is:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3}$$

At the start and the end points, the displacement constraint, speed constraint and acceleration constraint are expressed in (4)–(6), respectively:

$$\begin{cases} \theta(0) = \theta_0 \\ \theta(t_f) = \theta_f \end{cases} \tag{4}$$

$$\begin{cases} \theta'(0) = \theta'_0 \\ \theta'(t_f) = \theta'_f \end{cases} \tag{5}$$

$$\begin{cases} \theta''(0) = \theta''_0 \\ \theta''(t_f) = \theta''_f \end{cases} \tag{6}$$

Deriving Eq. (3), the velocity expression of the robot's trajectory is obtained as:

$$\theta'(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{7}$$

Similarly, by means of the second derivative of t in formula Eq. (6), we can get the acceleration function as follows:

$$\theta''(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{8}$$

Combining Eqs. (4), (5) and (6), we obtain the coefficients of the quintic polynomial as follows:

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \theta'_0 \\ a_2 = \frac{\theta''_0}{2} \\ a_3 = \frac{20\theta_f - 20\theta_0 - (12\theta'_0 + 8\theta'_f)t_f - (3\theta''_0 - \theta''_f)t_f^2}{2t_f^3} \\ a_4 = \frac{30\theta_f - 30\theta_0 + (16\theta'_0 + 14\theta'_f)t_f + (3\theta''_0 - 2\theta''_f)t_f^2}{2t_f^4} \\ a_5 = \frac{12\theta_f - 12\theta_0 - (6\theta'_0 + 6\theta'_f)t_f - (\theta''_0 - \theta''_f)t_f^2}{2t_f^5} \end{cases} \tag{9}$$

Introducing the above factors into Eq. (3), the robot trajectory equation of the quintic polynomial can be derived.

Improvement in genetic algorithm

The basic knowledge of the kinematics of the robot and the basic principle of the interpolation of the trajectory points of the joint space by the quintic polynomial have been introduced in the preceding narrative. On this basis, it is necessary to optimize the time interval of the interpolation point of the track point by using the genetic algorithm to realize the optimal planning of the robot time and further make the angular displacement, velocity and acceleration curve of each joint movement smoother.

The genetic algorithm, which is an artificial intelligence optimization algorithm for simulating the genetic and evolutionary processes in nature [10–12], owns the characteristics of simplicity and robustness, and starts from the parallel solution of the problem (rather than a single solution). Therefore, it not only has excellent global optimization but also is used in the optimization process of practical problems [13].

However, the crossover operator and mutation operator of a simple genetic algorithm are invariant in the process of algorithm implementation and do not satisfy the dynamic requirement of biological evolution. To this end, Ren and San [14] proposed an adaptive genetic algorithm according to the individual fitness of different dynamic adjustment of crossover probability and mutation probability. The adjustment of the crossover and mutation probabilities is shown in Eqs. (10) and (11), respectively:

$$p_c = \begin{cases} p_{c1} - \frac{(p_{c1} - p_{c2})(f' - f_{\text{avg}})}{f_{\max} - f_{\text{avg}}}, & f' \geq f_{\text{avg}} \\ p_{c1}, & f' < f_{\text{avg}} \end{cases} \tag{10}$$

$$p_m = \begin{cases} p_{m1} - \frac{(p_{m1} - p_{m2})(f' - f_{\text{avg}})}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ p_{m1}, & f' < f_{\text{avg}} \end{cases} \tag{11}$$

where $p_{c1} = 0.9$, $p_{c2} = 0.6$, $p_{m1} = 0.1$, $p_{m2} = 0.01$ f' is the parent of the larger fitness value; f_{avg} is the average fitness value in the population; $f_{\max}$ is the fitness value of the largest individual in the current population.

Equation (10) shows that a larger fixed crossover probability is given when the fitness value of the cross parent is small. And the greater the fitness of the two chromosomes, the smaller the crossover probability is. The adjustment of the mutation probability in Eq. (11) is consistent with the crossover probability.

It is noted that the above method has two drawbacks. Firstly, the crossover probability and the mutation probability are fixed for the parent chromosomes whose fitness is lower than the average fitness of the population. Secondly, the adjustment of crossover and mutation probability is linear, and it can not meet the objective of population evolution. To overcome the two drawbacks, this paper presents an adaptive genetic algorithm that uses a cosine function to adjust crossover probability and mutation probability. Adjusting Eqs. (10) and (11) as follows:

$$p_c = \begin{cases} \frac{p_{c0} + p_{c\min}}{2} + \frac{p_{c0} - p_{c\min}}{2} \cos\left(\frac{f - f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}\pi\right), & f \geq f_{\text{avg}} \\ \frac{p_{c0} + p_{c\max}}{2} + \frac{p_{c0} - p_{c\max}}{2} \cos\left(\frac{f_{\text{avg}} - f}{f_{\text{avg}}}\pi\right), & f < f_{\text{avg}} \end{cases} \tag{12}$$

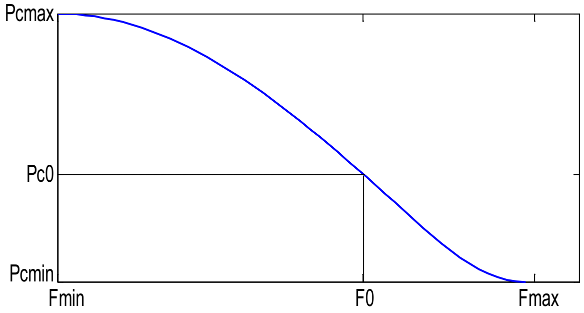

Fig. 3 Crossover probability adjustment curve of IAGA

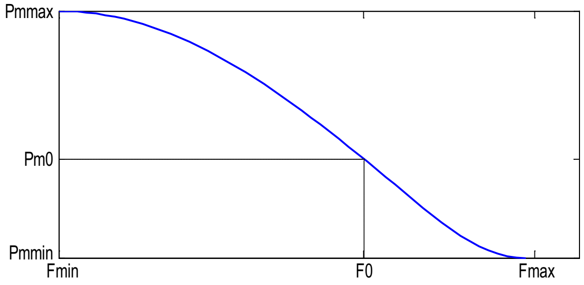

Fig. 4 Mutation probability adjustment curve of IAGA

$$p_m = \begin{cases} \frac{p_{m0}+p_{m\min}}{2} + \frac{p_{m0}-p_{m\min}}{2}\cos\left(\frac{f'-f_{\text{avg}}}{f_{\max}-f_{\text{avg}}}\pi\right), f' \geq f_{\text{avg}} \\ \frac{p_{m0}+p_{m\max}}{2} + \frac{p_{m0}-p_{m\max}}{2}\cos\left(\frac{f_{\text{avg}}-f'}{f_{\text{avg}}}\pi\right), \quad f' < f_{\text{avg}} \end{cases} \tag{13}$$

where $p_{c\min}$ and $P_{c\max}$ are the minimum and the maximum probability, respectively. $p_{c\min} \leq p_{c0} \leq p_{c\max}$ is a crossover probability; $p_{m\min}$ and $p_{m\max}$ denote the smallest and largest mutation probabilities, respectively, and $p_{c\min} \leq p_{m0} \leq p_{c\max}$ is a mutation probabilities; $f_{\max}$ is the fitness of the best individual in the population and f_{avg} is the average fitness; f is the greater fitness of crossed parent; f' is the fitness of the individual performing the mutation.

For the crossover probability adjustment method in Eq. (12), p_{c0} represents a crossover probability for the average fitness of the population, and the size of p_{c0} can be determined based on the required problem and the algorithm optimization process. If p_{c0} is larger, it means raising the crossover probability of the individual in the population and promoting the change in the individual gene pattern from a wide range. If x is small, the opposite is true. Therefore, in order to improve the optimization performance of the algorithm, we need to adjust the value of p_{c0} to balance the global optimization ability and local optimization ability of the algorithm.

According to the formula, the crossover probability and mutation probability of the improved algorithm can be approximately calculated, which are automatically adjusted according to the individual fitness in the population, as shown in Figs. 3 and 4.

The adaptive genetic algorithm is used to adjust crossover operator and mutation operator by the cosine function. The crossover probability and mutation probability are adjusted nonlinearly according to the fitness of the population. The algorithm flow is shown in Fig. 5.

Combining the improved algorithm with trajectory planning

Problem description

Assume that the robot performs an action with its end effector passing n points (including the start and the end points). N points can be converted into n corresponding joint variables of the joint space by the inverse kinematics of the robot, that is produced $n-1$ time segment with length of $T_i(i = 1, 2, 3, \ldots, n-1)$, $T_i = t_{i+1} - t_i$, where x represents the moment when the robot end effector reaches the i-th path point. The total time is:

$$T = T_1 + T_2 + \cdots + T_{i-1} = \sum_{i=1}^{n-1} T_i \tag{14}$$

where T is the total time of the robot movement which is the objective function of the problem; T_i $(i = 1, 2, 3 \ldots, n-1)$ is the time interval of the joint variable; the constraint is the maximum angular velocity, acceleration and jerk of the joints of the robot. Therefore, the problem of time optimal trajectory planning for robot is described as follows:

(1) Objective function:

$$\min \ T = \sum_{i=1}^{n-1} T_i \tag{15}$$

(2) constraint condition:

(A) angular velocity

$$\begin{cases} \forall t \in [t_i, t_{i+1}] \\ |\theta_i'(t)| \leq \theta_{\max}' \end{cases} \tag{16}$$

where $\theta_{\max}'$ is the maximum angular velocity allowed by the arm joint.

(B) acceleration

$$\begin{cases} \forall t \in [t_i, t_{i+1}] \\ |\theta_i''(t)| \ \leq \theta_{\max}'' \end{cases} \tag{17}$$

where $\theta_{\max}''$ is the maximum angular acceleration value allowed by the robot joint.

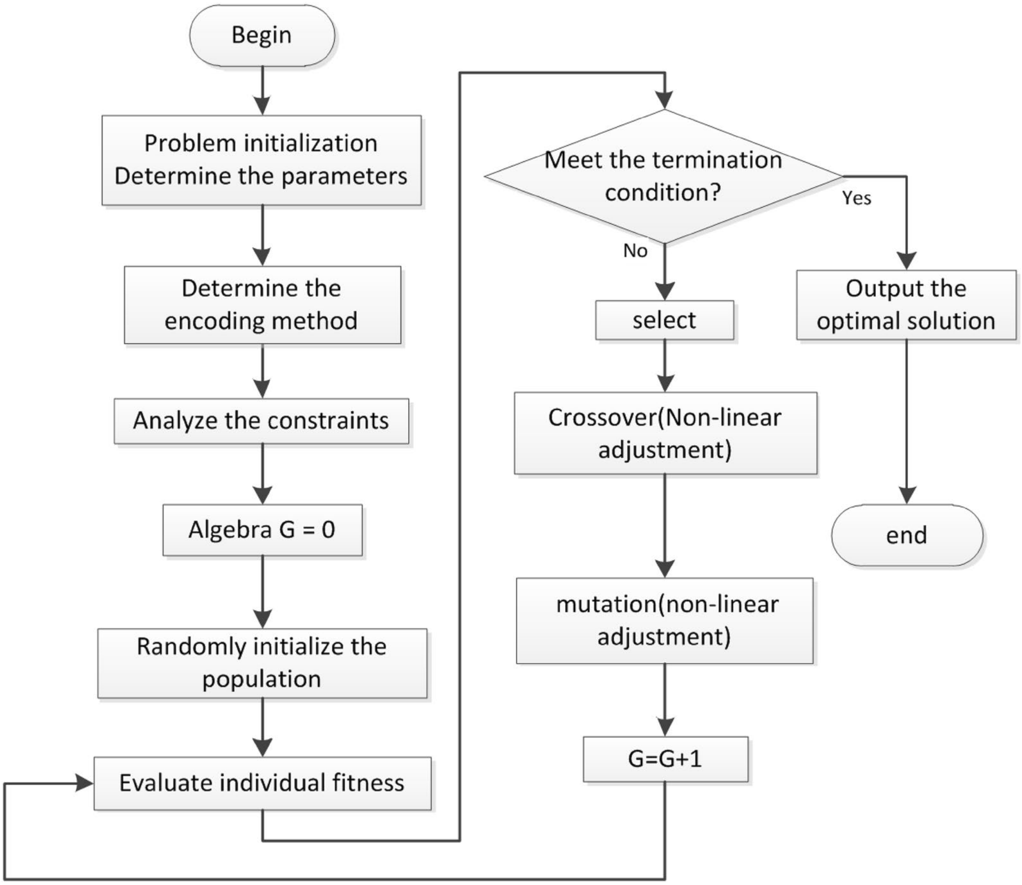

Fig. 5 Cosine genetic algorithm evolution process

(C) jerk

$$\begin{cases} \forall t \in [t_i, t_{i+1}] \\ |\theta_i''(t)| \ \le \theta''_{\max} \end{cases} \tag{18}$$

where $\theta'''_{\max}$ is the maximum jerk value allowed for the arm joint.

Time-optimal simulation

According to the parameters of the PUMA560 robot, we can get the constraints of the robot joints (see Table 1).

The improved algorithm is coded based on the real coding, and the parameters are selected as follows: population size $M = 80$; maximum crossover probability $P_{c1} = 0.9$, minimum value $P_{c2} = 0.4$, $P_{c0} = 0.7$; maximum mutation probability $P_{m1} = 0.1$, minimum value $P_{m2} = 0.01$, $P_{m0} = 0.7$; Evolutional generation $G = 100$. A MATLAB program for the optimal trajectory planning of the first three joints of PUMA560 is written by combining the quintic polynomial interpolation trajectory [15, 16]. In the optimization process, the trajectory of the robot joint is composed of the seven-segment polynomial curve, and its optimization precision is 0.001 s. The results are shown in Table 2.

It can be seen from Table 2 that after the optimization of the running time of the robot, the time taken by the robot to reach the target point is obviously reduced under the constraint of the joint angular velocity, acceleration and jerk of the robot. For the first three joints, the total time of the original run is 28 s. After optimization of this method, it is shortened to 13.729, 14.381

Table 1 The constraints of the robot joints

Joint *i*	Constraints		
	$\theta'(\mathbf{t})/(°/\mathbf{s})$	$\theta''(\mathbf{t})/(°/\mathbf{s}^2)$	$\theta'''(\mathbf{t})/(°/\mathbf{s}^3)$
1	100	45	60
2	95	40	60
3	100	75	55
4	150	70	70
5	130	90	75
6	110	80	70

Table 2 Optimization Results

Time interval	Joint *i*			
	Initial value	1	2	3
h_1	4	2.785	2.113	3.349
h_2	4	2.032	2.128	2.064
h_3	4	2.316	1.764	1.626
h_4	4	1.578	1.918	1.659
h_5	4	1.824	1.706	2.113
h_6	4	1.487	1.874	1.629
h_7	4	1.707	2.878	2.208
Total time (s)	28	13.729	14.381	14.648

and 14.648 s, which was at least 47.7% shorter than 28 s. Compared with several literatures that use PUMA560 as the object for MATLAB simulation, the same constraints apply. The reference literature [17] describes the method of interpolating the trajectory of the robot with seventh-order B-spline curves and optimizing the robot trajectory using the genetic algorithm. After optimization, the time is shortened from the original 20–15.620 s, which is shortened by 21.9%. In reference literature [18], a cubic B-spline curve is used to interpolate the robot motion trajectory, and then an improved genetic algorithm based on the crossover operator and mutation operator adjusted with evolutionary algebraic average fitness is used to perform the time optimal trajectory planning for the robot motion trajectory. The time after optimization was shortened from the original 20–13.072 s, a 34.6% reduction. It can be seen that the improved genetic algorithm, which the crossover operator and the mutation operator in the general adaptive genetic algorithm are adjusted cosine, based on quintic polynomial interpolation described in this paper satisfies the goal of the shortest time trajectory planning.

Simulation and analysis of smoothness of joint operation

The optimization of the robot's running time is shown in the previous section, and then, we simulate the angular displacement, velocity and acceleration curves of the first three joints of the robot.

From Figs. 6, 7 and 8, it can be seen that the angular displacement, velocity and acceleration curves are smooth and can reduce shock and impact of the robot arm and ensure smooth operation of the robot.

Conclusions

Based on the PUMA560 robot model, this paper briefly describes the method of locating the trajectory with the quintic polynomial interpolation in the joint space. Then, the crossover operator and the mutation operator in the general adaptive genetic algorithm are adjusted cosine to improve the performance of the algorithm. Using the improved algorithm to optimize the interpolation time of the robot trajectory, the simulations show that the

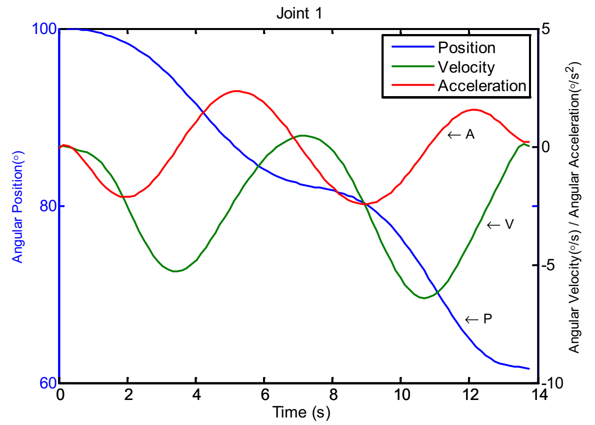

Fig. 6 Kinematic curve of joint 1

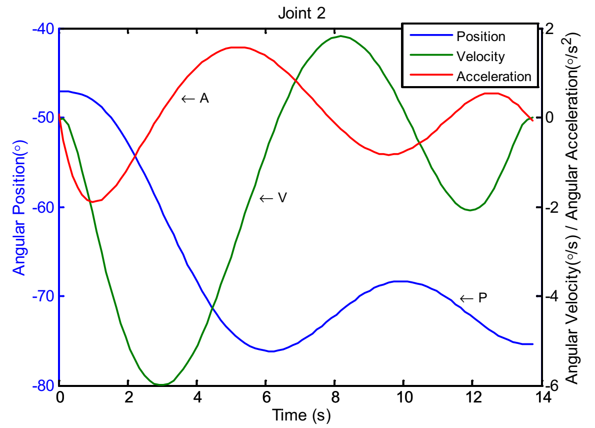

Fig. 7 Kinematic curve of joint 2

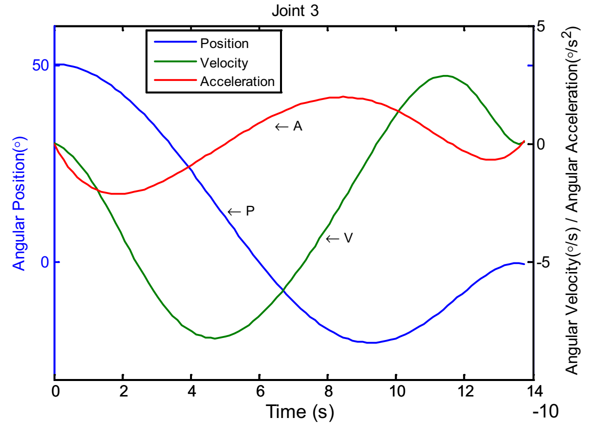

Fig. 8 Kinematic curve of joint 3

running time of each joint of the robot has been greatly reduced. The angular displacement, velocity and acceleration curve of the joint operation shows that the smoothness of the robot is better and the oscillations and shocks of the manipulator can be reduced. In conclusion, the method of this paper realizes the optimal trajectory planning target for robot time.

Authors' contributions

JYZ completed the system modeling and the main algorithm improvement. QXM worked on algorithm optimization and simulation. XGF and HS studied the improved method of adaptive genetic algorithm, analyzed the simulations and revised the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This research study was supported financially by the National Natural Science Foundation of China under Grant No. 61473171 and the Natural Science Foundation of Anhui Province (KJ2015A058).

Competing interests

The authors declare that they have no competing interests.

Ethics approval and consent to participate

Not applicable.

Funding

This research has been funded by the National Natural Science Foundation of China under Grant No. 61473171 and the Natural Science Foundation of Anhui Province (KJ2015A058).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 8 November 2017 Accepted: 14 May 2018
Published online: 03 July 2018

References

1. Kim KB, Kim BK. Minimum-time trajectory for three-wheeled omnidirectional mobile robots following a bounded-curvature path with a referenced heading profile. IEEE Trans Rob. 2011;27(4):800–8. https://doi.org/10.1109/tro.2011.2138490.
2. Chen H, Fang Y, Sun N, Qian Y (2015) Pseudospectral method based time optimal trajectory planning for double pendulum cranes. In: Control conference. IEEE; p. 4302–4307. https://doi.org/10.1109/chicc.2015.7260305.
3. Haili XU. Global time-energy optimal planning of industrial robot trajectories. J Mech Eng. 2010;46(9):19–25.
4. Liu H, Lai X, Wu W. Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Rob Comput Integr Manuf. 2013;29(2):309–17. https://doi.org/10.1016/j.rcim.2012.08.002.
5. Yang JT, Jiang WG, Lin YC. Jerk-optimal trajectory planning algorithm of industry robot. Sci Technol Eng. 2014;28:64–9. https://doi.org/10.3969/j.issn.1671-1815.2014.28.013.
6. Tohfeh F, Fakharian A (2015) Polynomial based optimal trajectory planning and obstacle avoidance for an omni-directional robot. In: Ai and robotics, vol. 2010. IEEE; p. 1–6. https://doi.org/10.1109/rios.2015.7270731.
7. Bende V, Pathak PM, Dixit KS, Harsha SP. Energy optimal trajectory planning of an underwater robot using a genetic algorithm. Proc Inst Mech Eng Part I J Syst Control Eng. 2012;226(8):1077–87. https://doi.org/10.1177/0959651812447232.
8. Zhu S, Wang. Time-optimal and jerk-continuous trajectory planning algorithm for manipulators. J Mech Eng. 2010;46(3):456–62. https://doi.org/10.3901/jme.20.
9. Guan-zheng Tan, Sheng-yuan HU. Real-time accurate hand path tracking and joint trajectory planning for industrial robots (I). J Cent South Univ Technol. 2002;9(4):273–8. https://doi.org/10.1007/s11771-002-0041-z.
10. Xu D, Liu M, Zhu L (2013) Single frequency GNSS integer ambiguity resolution with adaptive genetic algorithm. In: International conference on information science and technology. IEEE; p. 1049–1051. https://doi.org/10.1109/icist.2013.6747716.
11. Ling Wang. Intelligent optimization algorithm and its application. Beijing: Tsinghua University Press; 2001.
12. Liang Xu. Modern intelligent optimization hybrid algorithm and its application. 2nd ed. Beijing: Publishing House of electronics industry, Beijing; 2014 **(In Chinese)**.
13. Ying-Ying YU, Yan C, Tao-Ying LI. Improved genetic algorithm for solving tsp. Control Decis. 2014;29(8):1483–8. https://doi.org/10.13195/j.kzyjc.2013.0598.
14. Ren ZW, San Y. Improved adaptive genetic algorithm and its application research in parameter identification. J Syst Simul. 2006;18(1):40–1. https://doi.org/10.16182/j.cnki.joss.2006.01.011.
15. Yingjie L, Shanwen Z, Xuwu L et al (2005) MATLAB genetic algorithm toolbox and its application. Xi'an Electronic and Science University press, Xi'an **(In Chinese)**.
16. Jinwu Zhuo. The application of MATLAB in mathematical modeling. Beijing: Beijing Aerospace publishing house; 2011.
17. He J, Zhu L, Cheng L, Yin J (2015) Time-optimal trajectory planning of 6-dof robot based on genetic algorithm. J Mech Transm. https://doi.org/10.16578/j.issn.1004.2539.2015.09.010.
18. Niu Y (2013) Time-optimal trajectory planning of 6DOF serial robot. Dissertation, Changchun University of Technology, Changchun, Jilin, China.