Acta Technica *61*, No. 4A/2016, 189–200

Research on time optimal trajectory planning of 7-DOF manipulator based on genetic algorithm

Jianrong Bu[1], Junyan Xu[1]

Abstract. Trajectory planning is the basis of robot control. A robot with seven degrees of freedom is the object of study. The shortest time is taken as the trajectory optimization target. The joint trajectory planning is carried out by using B-spline curve. Under the constraint of kinematics parameters, the time optimization of the motion trajectory is realized by using the improved genetic algorithm. The results show that the total time of manipulator is reduced obviously, and the motion trajectory of each robot joint is continuous after optimization, which verifies the effectiveness and practicability of the algorithm.

Key words. 7-DOF, genetic algorithm, time optimization, trajectory planning.

1. Introduction

Trajectory planning is the movement trajectory of the robot in the process of movement, that is, the curve of the movement of the displacement, velocity, acceleration of the joints changing with time. The trajectory planning of robot is the basis of robot control and represents the research hotspot in the field of robotics in recent years. There are many performance indexes of trajectory planning, including time optimal trajectory planning, energy optimal trajectory planning, impact optimal trajectory planning and mixture optimal trajectory planning. Among them, time optimal trajectory planning is of great significance to improve the working efficiency of robots, and has been the focus of robot trajectory planning research.

The time optimal trajectory planning problem belongs to the nonlinear dynamic optimization problem. Its constraint conditions are generally nonlinear, the calculation process is complicated and the computation time is very long. For solving optimization of such problem, intelligent optimization algorithm are developed. The genetic algorithm has a good global searching ability and the advantage of inherent parallelism, so it has higher computational efficiency. In this study, the basic genetic

[1]Zhejiang Industry Polytechnic College, Shaoxing, 312000, China

algorithm is improved, and a time optimal trajectory planning method based on the B-spline technique is established.

2. Research overview of optimal trajectory planning

There are many different methods for time optimal trajectory planning of robots at home and abroad [1–5], such as quadratic programming method, dynamic target programming method, PID control method and iterative method. Liu et al. [6] proposed 3 -5 -3 spline function for the trajectory planning of the robot, which effectively avoids the vibration of the acceleration. Zhu et al. [7] used the B-spline curve interpolation method to construct the planned trajectory, and used the sequential quadratic programming method to solve the optimal moving time node, and then the time-optimal continuous trajectory was formulated that met the constraint of the nonlinearity kinematics. However, the interpolation time of polynomials is difficult to be optimized by traditional methods.

In the process of the conventional trajectory planning, the angular velocity, angular acceleration of the joint and joint torque and other variables generally usually take a relatively conservative value, so that it works under the rated value to avoid the robot beyond its maximum load capacity. Although this can ensure the safe and reliable operation of the robot, but this method can not make the robot fully perform its performance and its working efficiency is relatively low, because it does not consider the dynamic characteristics of the robot.

GUOTY uses PSO algorithm for the optimal trajectory planning of the space robot under the dynamics constraints. Based on the consideration of joint velocity, acceleration and acceleration constraints, Karami et al. [8] used genetic algorithm to plan the running time interval of each key point in joint space. However, there are some problems such as slow convergence rate, low accuracy, and local optimality in simple genetic algorithm.

We proposed a time-optimal trajectory planning method based on the improved genetic algorithm, which takes the optimal time in the working process of the robot as the objective function, considering the path constraints, the maximum angular velocity, the maximum angular acceleration and the maximum joint torque as a constraint to complete the task of trajectory planning.

3. Time optimal trajectory planning of seven-DOF manipulator based on genetic algorithm

3.1. Problem description

When the genetic algorithm is used to describe the optimization problem, the first thing to be solved is to encode the individual in the real problem, which is the basis of selecting, crossover and mutation in the process of genetic operation. In the case of a population of N individuals, this population needs to be given at the beginning of the genetic algorithm. In general, the value of the initial population object is given randomly. The population size N value generally takes 20 to 100, when N is large, the diversity of individuals in the population may increase. The initial population is equivalent to the given initial state in the genetic algorithm. Starting from the initial state, the solution is gradually optimized through the screening of the fitness function and the various operations of the genetic operator.

The genetic algorithm is a general method to solve the optimization problem. The design flow is basically the same for different optimization problems. The only difference is the parameter setting of the specific problem (including coding mode, genetic operator design, running competition value etc.). The general genetic algorithm design process is shown in Fig. 1.

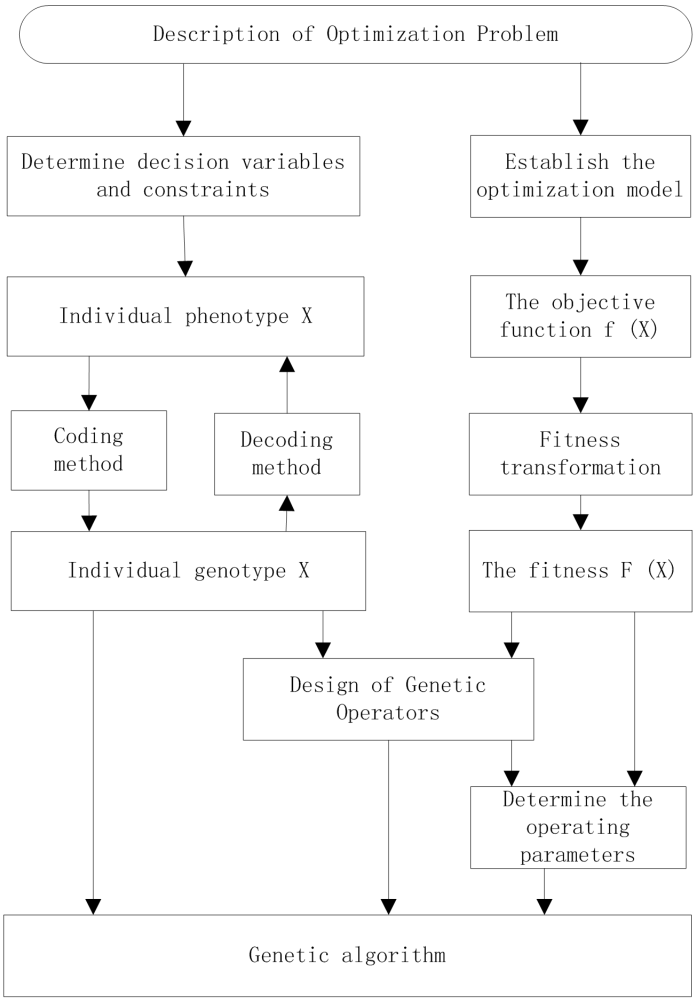

Fig. 1. Genetic algorithm design process diagram

3.2. Genetic algorithm optimization mathematical model

The genetic algorithm based on B-spline time optimization mainly includes the establishment of mathematical model of optimization problem and the design of genetic algorithm parameters. The general time optimization is carried out in the joint angular velocity and angle constraints, the advantages of this is less constraint, and the calculation is relatively simple, but the constraints of the joint torque is not considered, which may lead to large joint torque when the angular acceleration is large, so the actual capabilities cannot be realized. In order to solve this problem, the recurrence Newton-Euler algorithm is used to solve the joint torque, and the time optimal programming method under the constraint of joint torque is considered.

The mathematical model of genetic algorithm optimization problem is established, whose objective function is

$$T = \min \sum X_i \ \ (i = 1, 2, \cdots, m-1)\ , \tag{1}$$

where T is the total time of trajectory planning, m is the number of data points, and X_i is the time of the ith trajectory. In (1), X_i is an independent variable, which is chosen as the decision variable. In the study, B-spline programming is adopted as the constraint condition.

Assuming that the joint angular velocity in ith segment, $i = 1,\ 2,\ \cdots,\ m-1)$ is $\dot{q}_i\,(u)$. According to the B-spline in the three average B-spline trajectory plannings we can get

$$\dot{q}_i\,(u) = \frac{\mathrm{d}q}{\mathrm{d}t} = \frac{\mathrm{d}q/\,\mathrm{d}u}{\mathrm{d}t/\,\mathrm{d}u} = \frac{a_1u^2 + a_2u + a_3}{b_1u^2 + b_2u + b_3}\,, \tag{2}$$

where

$$a_1 = \frac{-q_{i-1} + 3q_i - 3q_{i+1} + q_{i+2}}{2}, \quad a_2 = q_{i-1} - 2q_i + q_{i+1}, \quad a_3 = \frac{-q_{i-1} + q_{i+1}}{2}\,,$$

$$b_1 = \frac{-t_{i-1} + 3t_i - 3t_{i+1} + t_{i+2}}{2}, \quad b_2 = t_{i-1} - 2t_i + t_{i+1}, \quad b_3 = \frac{-t_{i-1} + t_{i+1}}{2}\,.$$

The joint angular velocity in the ith segment obtained through (2) is $\dot{q}_i\,(u)$. For the maximum value $\dot{q}_i\,(u)_{\max}$ of the $\dot{q}_i\,(u)$, the $\dot{q}_i\,(u)$ can be obtained, respectively, through selecting limited discrete points in the section $u \in [0, 1)$, and $\dot{q}_i\,(u)_{\max}$ is approximated by the larger one to establish the speed constraint equation

$$\dot{q}_i\,(u)_{\max} \le \Theta\,, \tag{3}$$

where Θ is the maximum angular velocity allowed by joints.

Assuming that the joint angular acceleration in ith segment is $\ddot{q}_i\,(u)$, then

$$\ddot{q}_i\,(u) = \frac{\mathrm{d}\dot{q}}{\mathrm{d}t} = \frac{\mathrm{d}\dot{q}_i\,(u)\,/\,\mathrm{d}u}{\mathrm{d}t/\,\mathrm{d}u} = \frac{q^{''}t^{'} - q^{'}t^{''}}{t^{'3}}\,, \tag{4}$$

where

$$
\begin{aligned}
&q^{'} = a_1u^2 + a_2u + a_3,\\
&t^{'} = b_1u^2 + b_2u + b_3,\\
&t^{''} = 2b_1u + b_2 = c_1u + c_2,\\
&c_1 = 2b_1,\\
&c_2 = b_2,\\
&\ddot{q} = \left(-q_{i-1} + 3q_i - 3q_{i+1} + q_{i+2}\right)u + q_{i-1} - 2q_1 + q_{i+1} = d_1u + d_2,\\
&d_1 = -q_{i-1} + 3q_i - 3q_{i+1} + q_{i+2},\\
&d_2 = q_{i-1} - 2q_i + q_{i+1}\,.
\end{aligned}
$$

In this way we obtain

$$\ddot{\theta}\left(u\right) = \frac{\left(d_1u + d_2\right)\left(b_1u^2 + b_2u + b_3\right) - \left(c_1u + c_2\right)\left(a_1u^2 + a_2u + a_3\right)}{\left(b_1u^2 + b_2u + b_3\right)^3}\,. \tag{5}$$

The method is the same as getting the maximum angular velocity. The value of $\ddot{q}_i\left(u\right)_{\max}$ is approximated by getting the maximum of the limited discrete point in $u[0,1)$ section. The constraint condition of the acceleration is established as follows

$$\ddot{q}_i\left(u\right)_{\max} \le \dot{\Theta}\,, \tag{6}$$

where Θ is the maximum angular velocity allowed by joints.

In the constraint condition of the torque, according to the inverse kinetic Newton-Euler algorithm, the joint torque can be obtained according to the angular acceleration, angular velocity and angle of the joint at a certain moment. The time of the track passing through the ith segment is continuous. Therefore, in order to find the maximum joint moment during the ith segment, we need to discretize it and get the joint torque of a finite point. The maximum value of the joint torque is approximated as the maximum joint moment. The constraint condition of the joint torque is

$$\tau_{\max} \le \tau_{\mathrm{a}}\,, \tag{7}$$

where τ_{a} is the maximum torque allowed by joints.

4. Genetic algorithm design

In the task of designing genetic algorithm suitable for the optimization problem, the appropriate operating parameters are selected to complete the optimization of genetic algorithm. We first need to encode the chromosome. The choice of chromosome coding method is based on the constraints of this optimization problem. Because the inverse kinematics algorithm is used in the joint torque constraint, it needs to be called repeatedly. The constraint condition is complex and the computation requires a large capacity. Therefore, the floating-point coding method is suitable for solving the optimization problem.

The population size is always 20–100. In this algorithm, 20, 50, and 100 individuals are used to analyze the optimization results and verify the effect of population size on the precision of the algorithm. For the initial population value, the random variable between [30–50] is taken as the initial population (this is a very safe value according to the constraint of angular velocity and angular acceleration). In order to guarantee the accuracy of the algorithm, different interval is taken as the initial population to optimize the optimal solution. For the fitness function of individual fitness, we use Rank Fitness scaling in Matlab Genetic Algorithm Toolbox. The main principle is that the original fitness of the individual is calculated firstly, then the individual fitness ranges from large to small values $1, 2, \cdots, n$, and the new fitness of each individual is $\sqrt{1/n}$. The advantage is that the expansion effect of the original fitness is eliminated, so that the new fitness will be more concentrated, and it can reduce the probability of "premature" phenomenon.

The design of the selection operator is based on the "Stochastic uniform" method in the Matlab genetic algorithm toolbox. The principle of the algorithm is to draw a line in the proportion to the fitness of the dataset after the normalization. Then, the line moves according to certain criteria, and each time it stops the algorithm picks an individual dataset. Each time the individual that inherits to next generation accounted 0.2 of the population size. Mutation operator and crossover operator use the default operator in the toolbox, and the crossover probability is 0.8.

The above completes the genetic algorithm design for the optimization problem. It should be noted that there are more variables in genetic algorithms, and many variables are based on actual experience to determine, therefore, in the actual optimization process, it is necessary to repeatedly test. A variety of different parameters should be tested until you find the optimal solution. The flow chart of the genetic algorithm for optimization is shown in Fig. 2.

5. Results

The time optimal genetic algorithm proposed in this paper is validated by simulation, and the effect of optimization parameters and different parameters in the algorithm on the precision of the algorithm are analyzed. In the case of the same parameters, different population size and initial population were set respectively, and their effects on the efficiency and accuracy of genetic algorithm were compared. Under the premise that the initial population take the random number in interval [20–30], 20, 50 and 100 are respectively taken as the population size to compare the different optimization results, as shown in Figs. 3–6.

Figure 3 depicts the optimization process when the population size of genetic algorithm is 20. It can be found that when the optimization to the seventh generation meets the constraints of evolutionary condition, the evolution stops. In the figure, the dashed line represents the average value of objective function of each generation individual in the process of the evolution, and realizes the objective function value that represents the optimal individual. From the figure, we can see that as the evolution proceeds, the mean value gradually approaches the optimal solution. Finally, Fig. 4 shows the value of the fitness function when the evolution stops.

Figures 5 and 6 show individual values of optimal solution and evolution of pop-

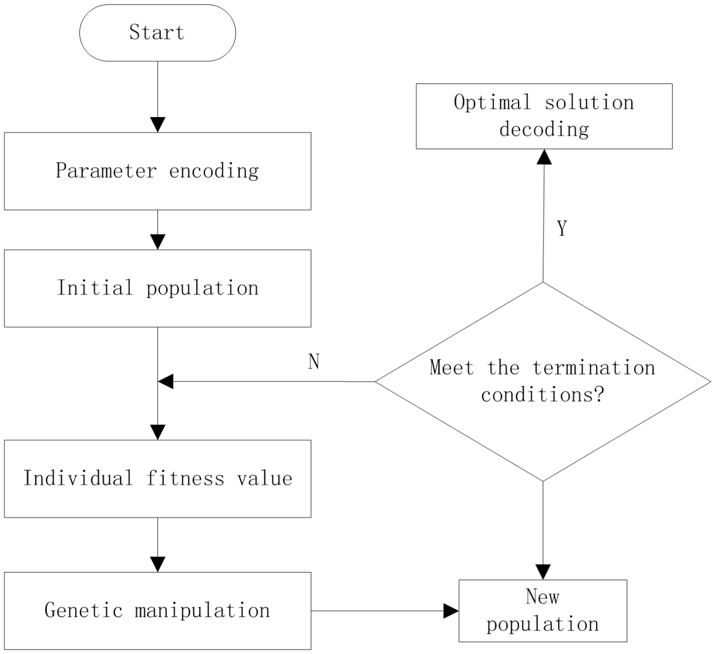

Fig. 2. Flow chart of genetic algorithm design

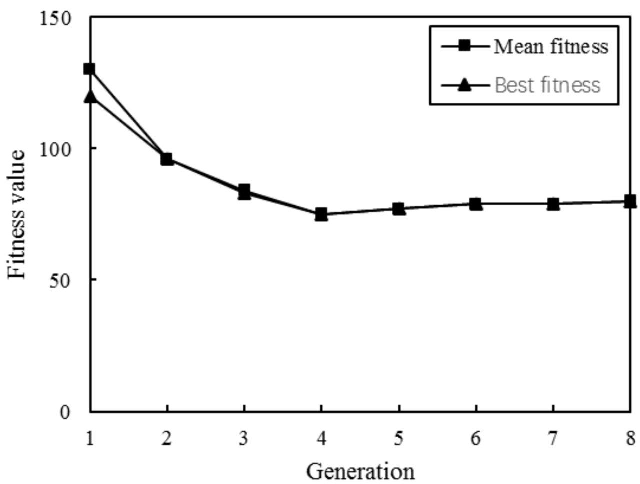

Fig. 3. Effect of population size on genetic algorithm: population size is 20

ulation size at 20, 50 and 100. From Fig. 5 we can see that the evolution process can approach the final optimal solution quickly when the population size is large. However, the different population sizes are optimized to the same result, finally.

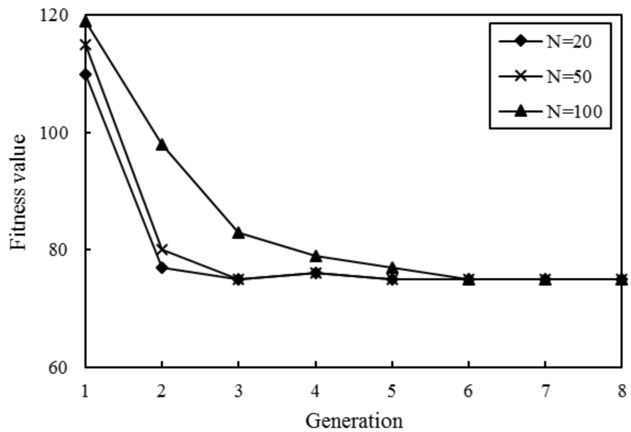

Fig. 4. Effect of population size on genetic algorithm: fitness value when the evolution stops

When the size of the population is large, the population diversity is better and the algorithm is more accurate. But the computational efficiency is sacrificed. In order to balance the two cases in the optimization process, the population size was selected to be 50.

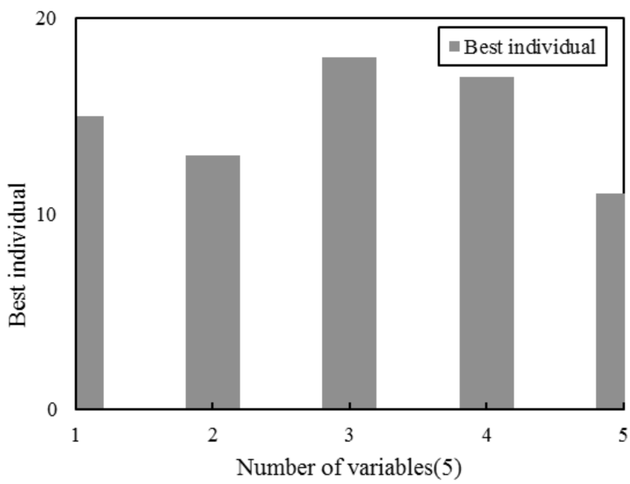

Fig. 5. Effect of population size on genetic algorithm: best individuals

In the genetic algorithm, the initial population value range is [20–30], which is a larger value that meets the condition based on the constraint of the speed and angular velocity. In order to verify the effects of data range of the initial population on the algorithm, the following groups of random number within range are taken respectively to carry out the simulation test, and the section range, respectively, is

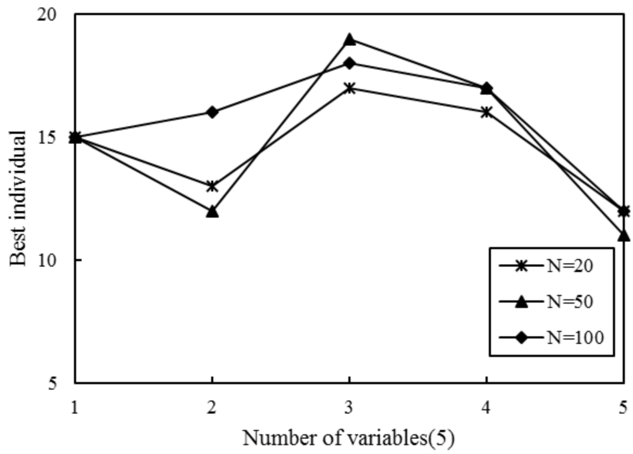

Fig. 6. Effect of population size on genetic algorithm: evolution of population size at 20, 50 and 100 individuals

[20–30], [30–40], [40–50], [20–50] and [10–30], and the population size is 50. The optimization of different interval is shown in Figs. 7 and 8. Figure 7 shows the variation of each generation and Fig. 8 shows the optimal values of different individuals in different intervals.

The results of optimization are summarized in Table 1.

Table 1. Effect of population size on genetic algorithm

Population size	X_1	X_2	X_3	X_4	X_5	T
$N = 20$	14.40	13.81	17.22	15.79	11.40	72.69
$N = 50$	14.20	12.92	18.21	16.45	9.90	71.67
$N = 100$	14.81	14.89	16.89	15.23	10.00	72.09

6. Discussion

In the study of seven-degree-of-freedom manipulator based on genetic algorithm, a seven-degree-of-freedom robot is used as the object of study. The trajectory is constructed by B-spline curves. Taking the shortest time as the optimization target, the coding scheme, selecting operator, crossover probability and mutation probability of the traditional genetic algorithm are improved for the simulation experiment of the trajectory optimization.

From the result analysis we can see that the process of evolution evolves to the optimal solution with the increase of the mean value of the range. For example, when the interval is [40–50], and the fourth generation is evolved, the individual

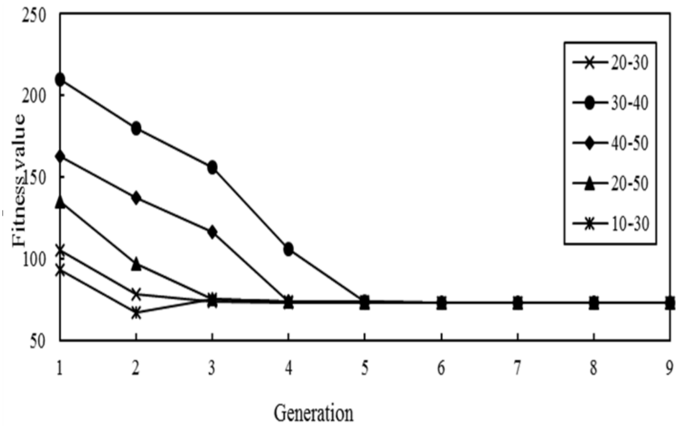

Fig. 7. Influence of initial population value on the algorithm: variation of each generation

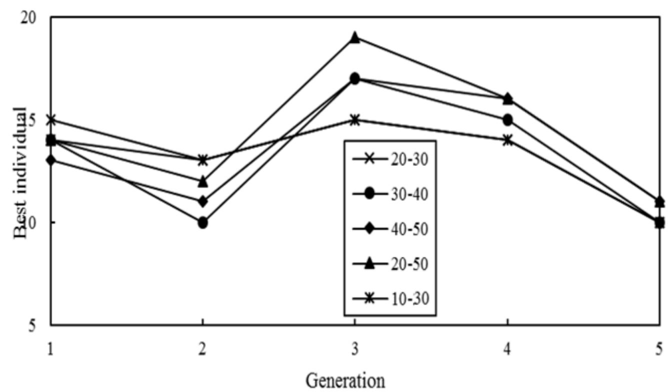

Fig. 8. Influence of initial population value on the algorithm: optimal values of different individuals in different intervals

converges to the optimal solution.. When the interval is [30–40], it will be close to the optimal solution when it evolves to the third generation, and only needs to evolve to two generations to reach the optimum when the interval is [20–30]. Interval [20–30] is closer to the optimal solution than the intervals [30–40] and [40–50], and when the interval [10–30] is taken into account, it can be seen from the figure that the solution of the evolution to the first generation is smaller than that of the later generations, so the individual values of the initial interval can be described as small. From the previous analysis, we can see that in the optimization problem, the initial value interval taken as [20–30] can not only guarantee the accuracy of the algorithm, but also improve its computational efficiency. The experimental results show that the manipulator's movement time is optimized from 107.52 s to 70.03 s by optimizing the genetic algorithm, which improves the operating efficiency of the manipulator and optimization of the trajectory time.

7. Conclusion

In this study, a genetic algorithm optimization model is established for the time optimal trajectory planning problem based on B-splines. The objective function, decision variables and constraints are determined. Constraints of angular velocity, angular acceleration and joint torque are considered in the constraint condition, the chromosome encoding method, the initial population, evaluation function, genetic operators and other parameters in the genetic algorithm are established, and finally the algorithm is realized through Matlab genetic algorithm toolbox to verify the genetic algorithm can effectively optimize the B-spline curve. The simulation results of the trajectory optimization simulation show that the time optimal trajectory planning of the seven-DOF manipulator based on genetic algorithm achieves the goal of optimizing the trajectory time. The results show that the proposed method is effective and reliable. The changing curve of the angular velocity, acceleration and acceleration of the robot joints has no abrupt change, which proves that the proposed method is effective and reliable.

References

[1] E. Barnett, C. Gosselin: *Time-optimal trajectory planning of cable-driven parallel mechanisms for fully specified paths with G1-discontinuities.* Journal of Dynamic Systems, Measurement, and Control *137* (2015), No. 7, paper 071007.

[2] M. Yuan, B. Yao, X. Zhu: *An efficient computation algorithm for time optimal trajectory planning with physical constraints.* Proc. IEEE IC Information and Automation (ICIA), 8–10 Aug. 2015, Lijiang, Yunnan, China, 3100–3105.

[3] B. Li, K. Wang, Z. Shao: *Time-optimal trajectory planning for tractor-trailer vehicles via simultaneous dynamic optimization.* Proc. IEEE/RSJ IC Intelligent Robots and Systems (IROS), 28 Sept.–2 Oct. 2015, Hamburg, Germany, 3844–3849.

[4] A. Berger, I. Ioslovich, P. O. Gutman: *Time optimal trajectory planning with feedforward and friction compensation.* Proc. American Control Conference (ACC), 1–3 July 2015, Chicago, Illinois, USA, 4143–4148.

[5] Y. Kim, B. K. Kim: *Efficient time-optimal two-corner trajectory planning algorithm for differential-driven wheeled mobile robots with bounded motor control inputs.* Robotics and Autonomous Systems *64* (2015), 35–43.

[6] H. Liu, X. Lai, W. Wu: *Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints.* Robotics and Computer-Integrated Manufacturing *29* (2013), No. 2, 309–317.

[7] S. Zhu, X. Jin, B. Yao, Z. Pan: *Non-linear sliding mode control of the lower extremity exoskeleton based on human–robot cooperation.* IJ Advanced Robotic Systems *13* (2016), No. 5, paper 1729881416662788..

[8] A. H. Karami, M. Hasanzadeh: *An adaptive genetic algorithm for robot motion planning in 2D complex environments.* Computers and Electrical Engineering *43* (2015), 317–329.

Received November 16, 2016