Acta Technica *62* (2017), No. 6B, 1279–1292

Position calculation method of robot obstacle avoidance based on ultrasonic sensor

Hang Xu[1]

Abstract. An obstacle avoidance path planning algorithm based on genetic algorithm has been proposed for the problem that the modular manipulator possibly collides with obstacle in the workspace. Firstly, the D-H representation is used for modeling of the mobile manipulator, and the kinematics and dynamics are analyzed; then, the kinematics and dynamics equations of the manipulator are established. Next, the basic content of the genetic algorithm is introduced. Then, under the comprehensive consideration, the time for movement of manipulator, the spatial distance of movement and the path length all shall be caused to be minimum, and simultaneously not to exceed the maximum setting torque of joint and any other conditions. The trajectory design has been optimized, and the process of genetic algorithm analyzed. Finally, through the experiment, the methods used are compared and verified in a single/multi obstacle environment(s), and the experimental results show that the corresponding constraints have been satisfied, and the position, velocity and acceleration curves of the trajectory are smooth and continuous. The simulation results show that the method is feasible, universal and effective, and possibly improves the efficiency of obstacle avoidance path planning.

Key words. Location calculation, Adjustment, Intelligent algorithm.

1. Introduction

In today's production and life, the mobile manipulator is playing an increasingly important role, and it has attracted the attention of scholars at home and abroad. The research focuses on the motion planning and control of mobile manipulator, the trajectory planning of which is the basis for trajectory control of the manipulator and of significance in stability, operating efficiency, operating accuracy and energy consumption of the manipulator. The obstacle avoidance planning means that a path from the starting point to the target point needs to be selected for the environmental

[1]Department of Mechanical & Automotive Engineering, Zhuhai College of Jilin University, Zhuhai Guangzhou, 519041, China

barrier, starting and target poses given, so that the moving object (robot) could safely go through all obstacles without any collision[1].

At present, the obstacle avoidance path planning methods for manipulator are various, mainly including C-space method, artificial potential field, visibility graph, pretreatment-planning algorithm, topological method and fuzzy logic algorithm, while these methods have disadvantages of complex modeling and huge calculation quantity. The Literature [2] presents an obstacle avoidance path planning algorithm based on A* algorithm that uses A* algorithm for collision-free path search in free working space of space manipulator to realize the obstacle avoidance path planning of space manipulator. In the Literature [3], the envelope of rule body is made available to the modeling of obstacle, and the idea on C-space method is used to map the obstacle and manipulator to two mutually perpendicular planes, so that the three-dimensional problem of manipulator working space is translated into a two-dimensional problem. In addition, the reverse optimization search method of two-branch tree is incorporated for path search, so as to realize the online real-time planning requirements. In the Literature [4], an obstacle avoidance planning algorithm of manipulator based on Q-learning dual neural networks has been presented, and this method has favorable average speed and success rate of target realization. For the Literature [5], through the development of linear inequality constraints, the internal points are used for solving the selection of optimization of optimal running speeds, and this method has no requirement for prior knowledge of movement of any objects in any environment and could effectively and online perform real-time collision avoidance action.

2. Establishment of MT-ARM manipulator model

The MT-ARM studied in this paper is a manipulator with four revolute joints and one gripper, which is shown in Fig. 1. The base coordinate system of the manipulator and the coordinate system of each joint have been established, which is shown in Fig. 2. The D-H parameters of the MT-ARM modular manipulator are shown in Table 1.

Table 1. D-H parameter table for MT-ARM

Joint No.	Coordinate system	θ_n	d_i/mm	$\alpha_n/\circ$
1	$O_0 - O_1$	$90° + \theta_1$	298.5	90
2	$O_1 - O_2$	$90° + \theta_2$	330	0
3	$O_2 - O_3$	$-90° + \theta_3$	0	-90
4	$O_3 - O_4$	0	320	0

3. Kinematics analysis of manipulator

Plane motion limits the motion of a robot in a single plane, and its computation is simpler than that of three-dimensional motion. However, most robot mechanisms

Fig. 1. MT-ARM

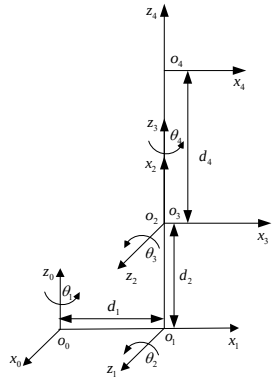

Fig. 2. D-H parameters coordinate system

can be reduced to plane mechanisms or processed based on the plane.

A pitching module is used as the last joint of the expandable modular manipulator of four degrees of freedom in this paper, which does not change the position of the end effector but changes. Thus MT-ARM modular manipulator model is simplified according to the geometric characteristics of manipulator, and the simplified plane manipulator of three degrees of freedom is shown in Fig. 3 where this manipulator is composed of a fixed connecting rod and three moveable connecting rods moving in the plane. All the connecting rods are connected with rotating countershaft, and the joint shaft is perpendicular to the connecting rod plane.

The specified length of connecting rod, namely the spacing between adjacent joint shafts are respectively expressed as L_1, L_2 and L_3. Order that point O, point A and B pointare respectively on the three joint shafts and point E is on the end effector. It is assumed that the connecting rod 1 fixed on the base connecting rod (connecting rod 0) is driven by effector 1 to form θ_1 and effector 2 drives the connecting rod 2 fixed on the connecting rod 1 to θ_2 form as the angle between two adjacent connecting rods the effector 3 drives the connecting rod 3 fixed on the connecting rod 2 to form θ_3. Since the manipulator execute tasks by moving the point E on the end effector, coordinate system xoy is introduced to describe the position, in which coordinate origin coincides with the first joint, and the position of the end effector is represented with coordinates (x_e, y_e). With the length of the connecting rod and the angle of the joint, the computational coordinates of the end effector are:

$$x_e = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)\,. \tag{1}$$

$$y_e = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)\,. \tag{2}$$

The orientation Φ_e of the end effector is indicated by the angle between the center line of the end effector and the forwardaxis,

$$\Phi_e = \theta_1 + \theta_2 + \theta_3\,. \tag{3}$$

In the event that the displacement value of the joint is given, the forward kinematics problem is to work out the coordinate and orientation of the end effector by means of formula (1), (2) and (3). The inverse kinematics problem is to obtain the corresponding displacement of the joint by the given orientation of the end effector.

As shown in Fig. 3, the coordinates x_e, y_e, and azimuth Φ_e of the end effector are known, and the angle θ_1, θ_2 and θ_3 of the three joints are required to be determined.

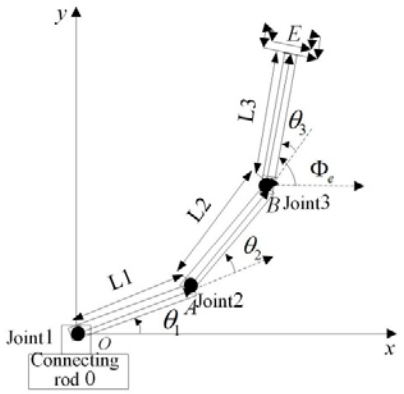

Fig. 3. Planar Robot of Three Degrees of Freedom

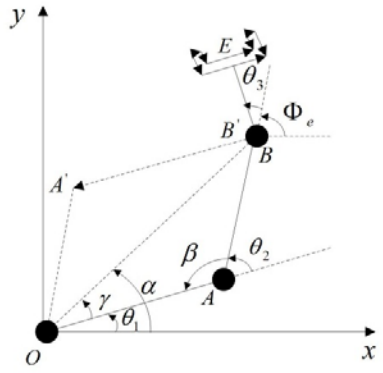

Fig. 4. Solution to the Inverse Kinematics of the Manipulator

Order that the coordinates of point B are x_w, y_w, then:

$$\begin{aligned} x_w &= x_e - L_3 \cos \Phi_e \,, \\ y_w &= y_e - L_3 \sin \Phi_e \,. \end{aligned} \tag{4}$$

Thus, $\alpha = \tan^{-1} \frac{y_w}{x_w}$.

It can be obtained as per the cosine theorem in the triangle OAB that $\beta = \cos^{-1} \frac{L_1^2 + L_2^2 - x_w^2 - y_w^2}{2L_1 L_2}$, $\gamma = \cos^{-1} \frac{x_w^2 + y_w^2 + L_1^2 - L_2^2}{2L_1 \sqrt{x_w^2 + y_w^2}}$ thus:

$$\begin{aligned} \theta_1 &= \alpha - \gamma \,, \\ \theta_2 &= \pi - \beta \,, \\ \theta_3 &= \Phi_e - \theta_1 - \theta_2 \,. \end{aligned} \tag{5}$$

Equation (5) defines a set of solutions of ,andand another set of solutions is shown simultaneously inand of Fig. 4, then:

$$\begin{aligned} \theta_1' &= \theta_1 + 2\gamma \,, \\ \theta_2' &= -\theta_2 \,, \\ \theta_3' &= \Phi_e - \theta_1' - \theta_2' = \theta_3 + 2\theta_2 - 2\gamma \,. \end{aligned} \tag{6}$$

4. Kinetic analysis of manipulator

In the statics of manipulator, the forces and torques acting on the manipulator are researched when the robot is stationary or in slow motion, while in the dynamics of the manipulator, the dynamic relation between the manipulator motion and the joint driving force (torque) is researched, and the differential equation describing this dynamic relation is called as the dynamic model.

The Lagrange equation of the particle system is used to analyze the dynamics issue of the manipulator. The Lagrange equation is:

$$L = K - P\,.$$

Where L represents the Lagrange function, K represents the sum of the kinetic energy of the system and P represents the sum of the potential energy of the system, then the kinetic equation is expressed as:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}\,. \tag{7}$$

Where τ_i represents the generalized force (force or torque) of joint i, $\dot{q}_i$ represents the generalized velocity ($\dot{\theta}_i$ or v), and q_i indicates the joint coordinates.

It is assumed that lengths of connecting rod of the three-link manipulator are respectively L_1, L_2 and L_3 the mass is respectively concentrated at m_1, m_2 an m_3, the coordinate system is selected as shown in Fig. 5.

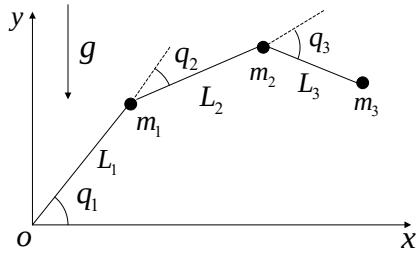

Fig. 5. Particle Coordinates of Three-link Manipulator

As shown in Fig. 5, the kinetic energy k_i and potential energy p_i at the particle $m_i (i = 1, 2, 3)$ are calculated respectively, and for the particle m_1, there are following equations:

$$\text{Kinetic energy:} \qquad k_1 = \frac{1}{2}m_1 v_1^2 = \frac{1}{2}m_1(L_1\dot{q}_1)^2 = \frac{1}{2}m_1 L_1^2 \dot{q}_1^2\,. \tag{8}$$

$$\text{Potential energy:} \qquad p_1 = m_1 g L_1 \sin(q_1)\,. \tag{9}$$

The coordinates of the particle m_2 are as follows:

$x_2 = L_1 \cos q_1 + L_2 \cos(q_1 + q_2)$

$y_2 = L_1 \sin q_1 + L_2 \sin(q_1 + q_2)$

The velocity component $\dot{x}_2$ and $\dot{y}_2$ are obtained as after derivation to x_2 and y_2, then $v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$.

$$\text{Kinetic energy:} \qquad k_2 = \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2)\,. \tag{10}$$

$$\text{Potential energy:} \qquad p_2 = m_2 g (L_1 \sin q_1 + L_2 \sin(q_1 + q_2))\,. \tag{11}$$

The coordinates of the particle m_3 are as follows:

$$\begin{aligned} x_3 &= L_1 \cos q_1 + L_2 \cos(q_1 + q_2) + L_3 \cos(q_1 + q_2 + q_3) \\ y_3 &= L_1 \sin q_1 + L_2 \sin(q_1 + q_2) + L_3 \sin(q_1 + q_2 + q_3) \end{aligned}$$

In a similar way, the following are be obtained:

$$\text{Kinetic energy:} \qquad k_3 = \frac{1}{2} m_3 v_3^2 = \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2)\,. \tag{12}$$

$$\text{Potential energy:} \qquad p_2 = m_3 g (L_1 \sin q_1 + L_2 \sin(q_1 + q_2) + L_3 \sin(q_1 + q_2 + q_3))\,. \tag{13}$$

Thus, we can obtain that $L = (k_1 + k_2 + k_3) - (p_1 + p_2 + p_3)$, substitute it into formula (7) to obtain the corresponding joint torque τ_1, τ_2 and τ_3 respectively.

5. Obstacle avoidance planning algorithm based on genetic algorithm

5.1. Overview of genetic algorithm

The genetic algorithm (GA) is a calculation method for simulation of biological genetic and evolutionary process, of which principle is to take advantage of biological evolution law of "survival of the fittest in natural selection". First of all, the parameters of the problem to be solved are encoded into chromosomes, and then the selection, crossover and mutation operations are performed to exchange the information of chromosomes in the population, and finally the chromosomes that satisfy the optimization target are generated.

In genetic algorithms, the corresponding data or arrays of chromosomes are generally represented by one-dimensional cluster-structured data, and each position on the cluster is corresponding to the value of gene. A chromosome is a gene cluster or genetic individuals. A population is made up of a certain number of individuals. Population scale is also the population size, which refers to the number of individuals in a population. Fitness (fitness) is the degree to which each individual is adapted to the environment.

First, several populations $P(t)$ are generated at random and the fitness value of the individual in the populations is calculated by means of the fitness function to evaluate each member. According to the principle that the chromosome with low fitness is inhibited while the one with high fitness is inherited, the chromosome corresponding to each individual is selected and then individual genetic operation is

performed by taking advantage of crossover and mutation operators to obtain a new generation of population.

Selection: according to certain rules, select some excellent individuals in population $P(t)$ of the generation t, and transfer these individuals to the next generation $P(t+1)$;

Crossover: match the individuals in the population $P(t)$ into a couple at random and exchange chromosomes between each couple of individuals with a crossover probability;

Mutation: change a allele value or some allele values of each individual in population $P(t)$ with the mutation probability.

After repeated operations in such way, the system evolves continuously toward a better solution, and eventually the resulting population satisfies the convergence condition and is most suitable for the problem environment, thus the optimal solution is obtained.

5.2. Implementation of genetic algorithm

The implementation of genetic algorithm mainly includes parameter coding, setting initial population, designing fitness function, genetic operation, setting algorithm control parameter and processing constraint condition.

(1) Code: the code of genetic algorithm is a genetic representation of solution, that is, the solution of the problem is encoded as the chromosome required for genetic algorithm, which can be divided into: gray code encoding and real number encoding, binary encoding and integer or alphabet encoding etc.;

(2)Initial population: in the range of the solution, a certain number of individuals are randomly generated, and each individual is generated by random initialization;

(3) Fitness function: as the basis of genetic manipulation and a criterion to evaluate the merits of each individual in a population, it reflects the natural law of "the fittest survive".

(4) election: according to the size of individual fitness value, the operation of survival of the fittest is adopted for each individual, so as to generate new population. As the number of generations in the breeding is less, the individuals with lower fitness are eliminated; the fitness value of the individuals with higher fitness is kept close to the optimal solution due to the large number of generations in the breeding. Selection operators include $(\mu+\lambda)$ selection, competitive selection, roulette selection, steady-state reproduction, array and scaling transformation and sharing.

(5) Crossover: it determines the global search ability of the genetic algorithm, and in some way, the paired chromosomes are exchanged with each other for partial genes to form two new individuals. The value range of general crossover probability P_c is $0.4 \sim 0.99$.

(6) Mutation: as auxiliary method of genetic algorithm to generate new individuals, it determines the global search ability of the genetic algorithm and changes a place value or some place values on the individual encoding string with a smaller probability to form the new individual. The value range of general mutation probability P_m is $0.0001 \sim 0.1$. .

(7) Termination of evolutionary generation: the genetic termination condition is usually taken as $100 \sim 500$ based on the complexity of the problem.

The main implementation steps of the genetic algorithm are shown in Fig. 6,

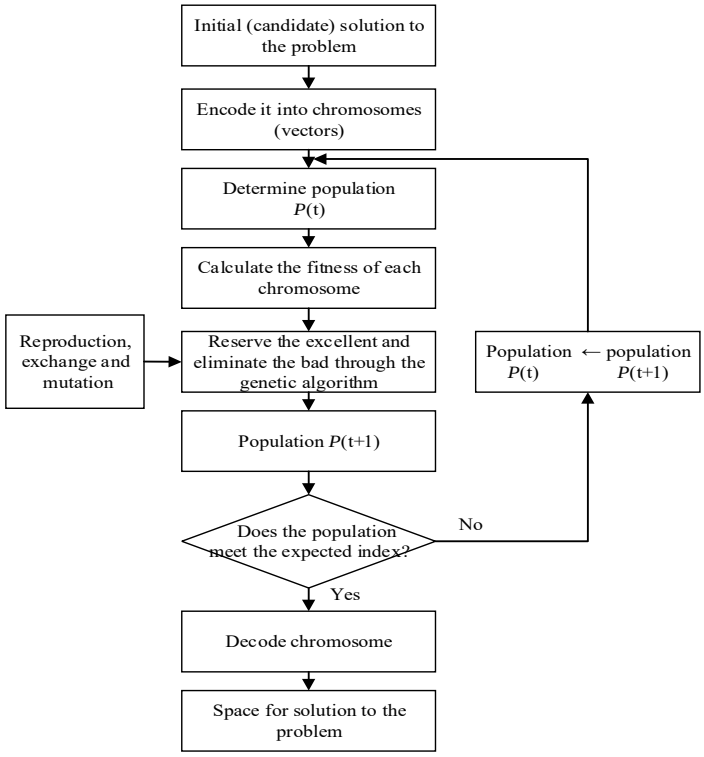

Fig. 6. Implementation Steps of Genetic Algorithm

Step1: Initialization: set the largest evolutionary generation number T and evolutionary generation number counter $t = 0$, randomly generate M individuals of initial population $P(t)$;

Step2: Individual evaluation: calculate the fitness value of each individual in the population $P(t)$;

Step3: Selection operation: exert the selection operator on the population;

Step4: Crossover operation: exert the crossover operator on the population;

Step5: Mutation operator: exert the mutation operator to the population, and the next generation of population $P(t+1)$ is obtained after selection, crossover and mutation operation of population $P(t)$;

Step6: Judgment of termination condition: when $t \leq T$, then go to step two; if $t > T$, terminate the calculation, and the optimal solution is the individual with the greatest fitness in the evolution process.

6. Experimental design and result analysis

6.1. Parameters setting of genetic algorithm

The quantity of selected population is 200, the coding scheme is real-number coding, $P_c = 0.8$, $P_m = 0.05$ and the fitness function is as follow:

$$F_f = \frac{1}{\beta_1 F_{ot} + \beta_2 F_q + \beta_3 F_c + \beta_4 t_T}. \tag{14}$$

Where, F_{ot} represents the penalty function of joint torque, F_q represents the total distance traveled by each joint, F_c represents the trajectory length of space, $t_T = t_1 + t_2$ indicates the total running time, and $\beta_i (i = 1, 2, 3, 4)$ means the weight coefficient.

$$F_\alpha = \sum_{j=1}^{b} \sum_{i=1}^{\alpha} F_i^j, F_i^j = \begin{cases} 0 & \tau_i^j < \tau_{i\max} \\ |\tau_i^j| - \tau_{\max} & \tau_i^j \geq \tau_{i\max} \end{cases} \tag{15}$$

Where, α represents the degree of freedom of manipulator, b represents the number of nodes between the starting point and the end point, $\tau_{i\max}$ indicates the maximum torque of the joint.

$$F_q = \sum_{i=1}^{a} \sum_{j=2}^{b} |q_{ij} - q_{ij-1}|. \tag{16}$$

$$F_c = \sum_{j=2}^{b} d(p_j, p_{j-1}) \tag{17}$$

Where, $d(p_j, p_{j-1})$ represents the distance between joint p_j and p_{j-1}.

When obstacles exist in the workspace, the objective function of obstacle avoidance is defined as F_{ob}, when the manipulator moves from the starting point to the end point, and the manipulator does not intersect the obstacle area, $F_{ob} = 1$, otherwise $F_{ob} = 0$, that is as follow:

$$F_{ob} = \begin{cases} 1, & \sum_{j=1}^{b} \sum_{i=1}^{a} (link_{ij} \cap obstacle) = 0 \\ 0, & \text{other} \end{cases} \tag{18}$$

At this time, the fitness function is defined as $F = F_{ob}/F_f$.

6.2. Parameters setting of manipulator

The connecting rod length of manipulator respectively are $L_1 = 1$, $L_2 = 1$, $L_3 = 0.5$, the weight coefficients respectively are $[\beta_1, \beta_2, \beta_3, \beta_4] = [2, 2, 2, 1]$. The maximum torque of three joints respectively are $\tau_{1\max} = 45Nm$, $\tau_{2\max} = 20Nm$, $\tau_{3\max} = 45Nm$, the speed and accelerated speed are zero at starting and ending

time, the constraint range of optimization objective $[t_1, t_2, f_q, f_c]$is as follow:

$$\begin{aligned} &1s \le t_i \le 8\,s \qquad (i = 1, 2) \\ &0 < f_q < +\infty \\ &2.1m \le f_c < +\infty \end{aligned}$$

6.3. Experimental results and analysis

Experiment I: Trajectory planning of manipulator in environment with single obstacle

As shown in Figure 7. Set the starting point of the manipulator is $(x = 0.5, y = 2, \Phi_e = 100°)$, the $(-2, 0.5)$ coordinates of end point is (0, 2), and the radius of obstacle is $r = 0.35$ m. Figure 7 (a) presents the trajectory curve for end effector of manipulator; Figure 7 (b), (c), (d) respectively shows movement time t_T, movement distance of joint f_q, and the curve of trajectory length changing with genetic algebra; Figure 7 (e), (f), (g), (h) respectively rep-resents the curve of the joint angle, velocity, acceleration and joint torque changing with time, the hollow circle represents the optimized intermediate node.

The experimental results show that the result of trajectory optimization is as follows: the total running time is $t_T = 4.0013\,s$, the movement distance of joint is $f_q = 3.4612m$, and the trajectory length is $f_c = 3.1343m$. Total overshoot torque is $f_{ot} = 0$.

Based on Figure 7 (b), (c), (d), the movement time t_T, joint movement distance f_q and f_c trajectory length are changed with the increased genetic algebra and tending to a stable value; according to Figure 7 (e), (f), (g), the joint angle, velocity and acceleration function express continuous curve; Figure 7 (h) shows that the torque of each joint is within the each allowable maximum range.

Experiment II: Trajectory planning of manipulator in environment with multiple obstacles

As shown in Figure 8. Set the starting point of the manipulator is $(x = 0.5, y = 2, \Phi_e = 100°)$, the coordinates of end point is $(-2, 0.5)$, the coordinates of obstacle is (0, 2), and the radius of obstacle is $r = 0.35m$, the coordinates of obstacle 2 is $(-1.3, 1.6)$, radius is $r = 0.21mm$, and the coordinates of obstacle 3 is $(-1.9, 1)$, radius is $r = 0.16m$. Figure 8 (a) presents the trajectory curve for end effector of manipulator; Figure 8 (b), (c), (d) respectively represents the curve of the joint angle, velocity, acceleration and joint torque changing with time, the hollow circle represents the optimized intermediate node.

The experimental results show that the result of trajectory optimization is as follows: the total running time is $t_T = 3.9325s$, the movement distance of joint is $f_q = 4.1246m$, and the trajectory length is $f_c = 3.1535m$. Total overshoot torque is $f_{ot} = 0$.

By Figure 8(a), the manipulator can move from the starting point to the end point in the environment with multiple obstacles, while avoiding the obstacles, based on Figure 8 (b), (c), (d), the movement time t_T, joint movement distance f_q and trajectory length f_c are changed with the increased genetic algebra and tending to a stable value; according to Figure 8 (e), (f), (g), the joint angle, velocity and

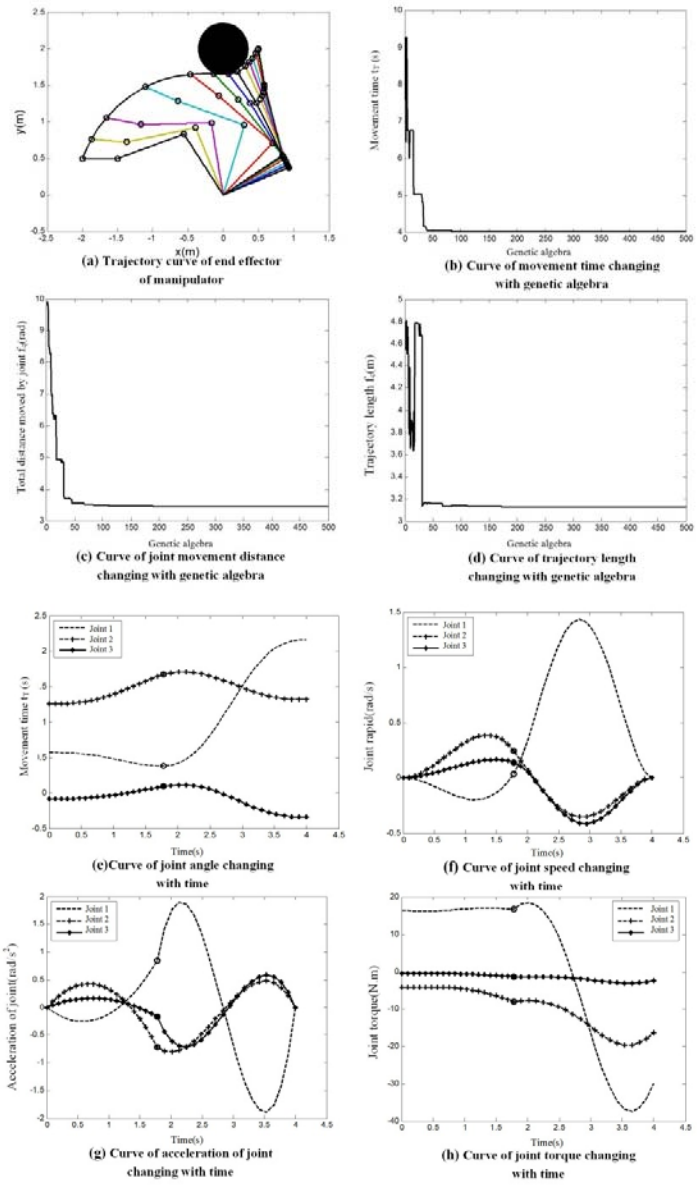

Fig. 7. Trajectory planning of manipulator in a single obstacle environment

acceleration function express continuous curve; Figure 8 (h) shows that the torque of each joint is within the each allowable maximum range.

From Experiment II and II, it can be seen that the trajectory length and total joint movement distance will increase following with the increase of the number of obstacles in the environment.

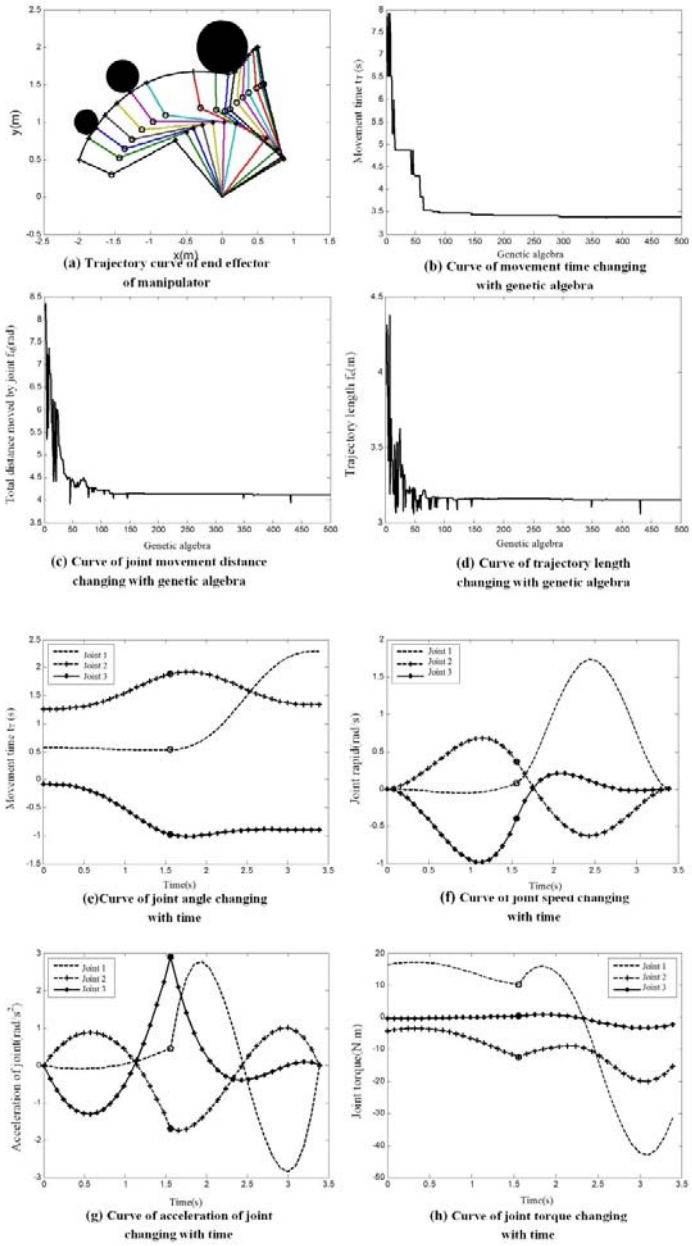

Fig. 8. Trajectory planning of manipulator in environment with multiple obstacles

7. Conclusion

This paper proposed an obstacle avoidance path planning method of manipulator based on genetic algorithm, to carry out the research in environment with single and multiple obstacles with optimized index of movement time, joint movement distance and trajectory length, the main content of research is obstacle avoidance path planning method of manipulator based on genetic algorithm, firstly, the D-H (Denavit-Hartenberg) representation is used for modeling of the mobile manipulator, and the kinematics and dynamics are analyzed; then, the kinematics and dynamics equations of the manipulator are established. On this basis, the optimization of obstacle avoidance path planning for manipulator is achieved by using genetic algorithm and taking movement time, moved space distance and trajectory length as optimized indexes in environment with single and multiple obstacles. Through simulation, the effectiveness and feasibility of the obstacle avoidance path planning algorithm based on genetic algorithm are verified, the algorithm improves the efficiency of the manipulator effectively avoiding the obstacles in the workspace.

Acknowledgement

Premier-Discipline Enhancement Scheme Supported by Zhuhai Government Fund (2015YXXK14).

References

[1] Nara S, Takahashi S: (2008) *Obstacle Avoidance Control for Mobile Robot Based on Single CCD Camera and Ultrasonic Sensors*[J]. Ieej Transactions on Sensors & Micromachines, 128:733-741.

[2] Pahlavan K, Krishnamurthy P and Geng Y: (2015) *Localization challenges for the emergence of the smart workd.* IEEE Access, 3(1) :3058-3067

[3] Ohya A, Kosaka A, Kak A: (1998) *Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing*[J]. IEEE Transactions on Robotics & Automation, 14(6):704-711 vol.2.

[4] Lv, Z., Tek, A., Da Silva, F., Empereur-Mot, C., Chavent, M., & Baaden, M.: (2013). *Game on, science-how video game technology may help biologists tackle visualization challenges.* PloS one, 8(3), e57990.

[5] Ohya A, Kosaka A, Kak A: (1997) *Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing*[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 1997:704-711 vol.2.

[6] Li S X, Yuan Q: (2014) *The Study of Robot Obstacle Avoidance Based on Fuzzy Control*[J]. Applied Mechanics & Materials, 577:386-389.

[7] Lv, Z., Halawani, A., Feng, S., Li, H., & Réhman, S. U.: (2014). *Multimodal hand and foot gesture interaction for handheld devices.* ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 11(1s), 10.

[8] Wang H, Zhao L Y, Chen W: (2014) *A Mobile Robot Obstacle Avoidance Method Based on Improved Potential Field Method*[J]. Applied Mechanics & Materials, 467:496-501.

[9] Weisen Pan, Shizhan Chen, Zhiyong Feng: (2013) *Automatic Clustering of Social*

Tag using Community Detection. Applied Mathematics & Information Sciences, 7(2): 675-681.

[10] Yingyue Zhang, Jennifer W. Chan, Alysha Moretti, and Kathryn E. Uhrich: (2015) *Designing Polymers with Sugar-based Advantages for Bioactive Delivery Applications*, Journal of Controlled Release, 219, 355-368.

[11] Cheng T T, Mahyuddin M N: (2009) *Implementation of behaviour-based mobile robot for obstacle avoidance using a single ultrasonic sensor*[C]// IEEE, 2009:244-248.

[12] Yingyue Zhang, Qi Li, William J. Welsh, Prabhas V. Moghe, and Kathryn E. Uhrich: (2016) *Micellar and Structural Stability of Nanoscale Amphiphilic Polymers: Implications for Anti-atherosclerotic Bioactivity, Biomaterials*, 84, 230-240.

[13] Gao L N, Deng J W, Xiao S Y: (2014) *Research on Obstacle Avoidance Behavior of Ultrasonic Sensor-Based Robot*[J]. Applied Mechanics & Materials, 457-458:778-782.

[14] Liu Q, Lu Y G, Xie C X: *Optimal Genetic Fuzzy Obstacle Avoidance Controller of Autonomous Mobile Robot Bas2006)ed on Ultrasonic Sensors*[C]// IEEE International Conference on Robotics and Biomimetics. IEEE, 2006:125-129.

[15] Jennifer W. Chan, Yingyue Zhang, and Kathryn E. Uhrich: (2015) *Amphiphilic Macromolecule Self-Assembled Monolayers Suppress Smooth Muscle Cell Proliferation*, Bioconjugate Chemistry, 26(7), 1359-1369.

Received May 7, 2017