

DYNAMIC PROGRAMMING PATH OF MOBILE ROBOT BASED ON EVENT-DRIVEN PROCESS

Z.Y. HUANG

*Shaoxing University, College of Yuanpei, China.
ElizabethxZX@yahoo.com*

Abstract: Event-driven process chain is an important method for path flow and system optimization in the process of robot dynamic planning path research. Path planning is an important part of robot research. The path planning problem and the path optimization planning problem of the robot in the dynamic environment are explored deeply in this paper based on the event-driven process chain perspective and the actual needs of the research project. Therefore, it uses the event-driven process chain method to optimize the path according to the original process and existing requirements based on the concept of event-driven process chain. A dynamic path planning model under event-driven process chain is constructed, and then the function, user and permission, code, input and output, and security of industrial robot system are optimized.

Keywords: genetic algorithm; mobile robot; planning path.

I. INTRODUCTION

Functions and events are the two most important elements that constitute the dynamic path planning model based on event-driven process chain perspective. Functions represent how an event occurs, and events represent a point of change in a state in the process. The basic idea of dynamic path planning for industrial robots from the perspective of event-driven process chain is to use static resources to form a dynamic model.

It embodies a value-added optimization process of industrial robot dynamic path planning. It has the characteristics of attributes, delay and randomness. Each element is given different attributes. Among them, the function objects are distributed randomly and the duration of each process can be calculated (Abu-Dakka *et al.*, 2016).

Path planning is to find a collision-free path from the initial state to the target state (Wang X *et al.* 2016) in an environment with obstacles according to certain evaluation criteria. The path planning problem of industrial robots in dynamic environment refers to the planning and control of the robot from the initial state to the target state in the expected planned way in the environment of robot, target and obstacle movement. The robot can not only make a quick stop to the target point, but also can track the moving target quickly, and can avoid obstacles (Kim *et al.*, 2016) effectively.

Path planning is mainly applied to the bottom strategy of robots, which is the basis of robot action. It

is found that path planning is the key problem to realize the control of industrial robot through the study of dynamic path planning of industrial robot. The results of robot dynamic path planning can affect the real-time and accuracy of robot action. The computational complexity and stability of the planning algorithm also affect the efficiency of the robot indirectly.

II. STATE OF THE ART

Path planning in dynamic environment has become an urgent problem to be solved with the development of robot technology. The difficulty lies in that sufficient environmental information is required and forming a closed loop with the environment, and requiring fast processing speed to meet real-time requirements. It needs to be planned in the state space, not only considering the factors of the robot itself, but also combining the surrounding dynamic environment and obstacles. The occurrence of dynamic obstacles has randomness and uncertainty. Moreover, path performance requirements during robot motion have multiple objectives, such as the shortest path, the best time, the best safety performance, and the lowest energy consumption, but there are often conflicts between them. Path planning and speed planning issues need to be addressed continuously (Pereira *et al.*, 2016).

The main methods of local planning include artificial potential field method, genetic algorithm and fuzzy logic algorithm. This method focuses on the local environmental information close to the distance robot, so that the robot has good collision avoidance ability (Liu *et al.*, 2017). Compared with the global planning method, the local planning method is more real-time and practical. The disadvantage is that relying only on local information, sometimes local poles are generated, and the robot cannot reach the destination smoothly (Zhao *et al.*, 2017). The method of distinguishing planning from the perspective of robot working environment can be divided into static determining environmental planning method and dynamic changing environmental planning method.

The planning problem in the dynamic environment has attracted people's attention and has achieved some results, which will be the development direction in the future (Wang *et al.*, 2017). The obstacle avoidance problem to be solved under the dynamic environment is studied in this paper, and finally the genetic algorithm-based behavior control method is used to solve the path planning problem.

III. METHODOLOGY

A. Mobile Robot Path Planning Based on Genetic Algorithm

The calculation of the fitness value can be regarded as the interface between the genetic algorithm and the optimization problem (Zhang *et al.*, 2016). The genetic algorithm evaluates the quality of a solution depends on the fitness value corresponding to the solution instead of the structure of its solution. Using the fitness value of each individual in the population to search, the selection of the fitness function will directly affect the convergence speed of the genetic algorithm and whether the optimal solution can be found or not. To carry out the path planning of industrial robots, it is necessary to plan a shortest feasible path (Montiel *et al.*, 2015). The constraint is that the path does not intersect the obstacle and requires a certain distance from the obstacle. The evaluation of the pros and cons of the path is used as the adaptive value of the chromosome in the genetic algorithm. The path is the shortest as the evaluation criteria, and the adaptive value function is selected as follows,

$$f = 1 / \left(\left(1 + \frac{1}{\sqrt{n-1}} \right) D \right) \quad (1)$$

Among them, n represents the total number of grids that the individual passed. D represents the sum of the linear distances between adjacent numbers of the individual. After defining the initialization population and the fitness value, the genetic algorithm can generate new individuals through genetic operators such as crossover and mutation. For genetic algorithms using floating-point coding, it is very important to define reasonable genetic operators to improve the performance of the algorithm. The offspring are generated by the evaluation of new individuals and evolved until they get the optimal solution to the problem or reach the algorithm termination condition (Yoon *et al.*, 2015).

A certain algorithm is used to select the same number of new generation groups from the current population. First, the fitness function of each individual is calculated. A method called "roulette" is adopted in this paper (Ran *et al.*, 2015). Set the size of the group as N . If the fitness of a certain body i is $f(x_i)_{(i=1, \Lambda, N)}$, the probability that the selected i -th chromosome is given by the following formula: the probability of being selected is expressed as follows.

$$P_i = F(x_i) / \sum_{j=1}^N F(x_j) \quad (2)$$

When selecting a chromosome, first turn the wheel, and when the turntable stops, the i corresponding to the grid pointed by the pointer is the selected individual. After N times of "roulette", a population of size N is obtained (Song B *et al.* 2016). Crossing is a certain probability and usually there are several methods such as single point crossing, multiple point

crossing, and uniform crossing. A turning point other than the starting point and the target point is randomly selected in the variable length chromosome, and the turning point is used as the intersection point, and the entire path is divided into two path segments. Then the two parent chromosomes are selected and the path segments behind the intersections are exchanged, thus producing two progeny chromosomes.

The crossed progeny chromosome replaces the parental chromosome of the original population, producing a new population. The mutation operator replaces a certain gene on a certain chromosome, and the local random search corresponding to the path planning is combined with the selection operation to ensure the validity of the genetic algorithm and make the genetic algorithm have local random search ability (Weerakoon *et al.*, 2015).

It is possible to select a starting point and an ending number as a variation point from the individual with a certain probability, delete the point, or replace it with another randomly generated serial number, or randomly select a serial number in the individual to insert at the variation. In the initial stage of evolution, the range of variation is large, but as the evolution progresses, the amount of variation gradually decreases.

In the path planning of the robot, special operations need to be added to ensure that the resulting path can avoid obstacles and meet the shortest requirements. Commonly, there are operations such as insert operator and delete operator. Insert operator: Since cross-operation and mutation operations may result in discontinuous paths, the purpose of introducing an insert operation is to compensate for the discontinuous path with a free raster, making it a continuous path. The specific method is to log through the obstacle segment for a path and record it.

If the wire segment crosses the obstacle, then a segment is selected from it and a point is added between the segments. Otherwise, a segment of the path will be selected and an arbitrarily generated turning point will be inserted, which requires the newly inserted turning point coordinates to be in the non-obstacle area. Obviously, there is no need to impose an insertion operation where there is no intersection, so the insertion operation is performed with unequal probability.

Delete operator: Inserting a free raster may result in duplicate nodes in the path. The function of the deletion operation is to round off the redundant sequence number between two identical serial numbers in the individual, together with one of the two identical serial numbers, so that the infeasible path becomes a feasible path to simplify the path.

B. Genetic algorithm for robot behavior control

The encoded object is the parameter of mobile robot behavior control. After analyzing the hardware resources and working environment of the robot

system, the four behaviors of the previous section are described by coding, using mathematical methods or mathematical models. This translates the optimization problem into a search space that can be processed with genetic algorithms. In order to prevent the occurrence of precocity (that is, prematurely falling into the local optimal solution), the population requirements have a certain scale, but the scale selection is too large, and the calculation process becomes more complicated.

According to the actual situation, the population size is selected between 10 and 200. So, the appropriate population size (`pop_size`) based on experience is chosen. Let N individuals were randomly generated between 10 and 200 as the initial population $P(0)$. In this paper, four parameters are optimized by genetic algorithm, so the population is a matrix, each of which is composed of four parameter groups, and the parameter group is $\lambda_i = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]$, $i \leq \text{pop_size}$. That is, the population can be expressed as $\text{pop_size} \times 4$. Each parameter in the individual of the population is a number randomly generated between (5, 100).

For the path planning of the movement of the mobile robot, it is required that the path does not intersect with the obstacle, and with a certain distance, the selection of the fitness function will directly affect the quality of the optimized parameters. Therefore, the design of fitness function and the collision between robot and static obstacle are related to the collision of dynamic obstacle, the maintenance of motion formation and the path of robot.

Therefore, the problem to be solved in this paper belongs to the multi-objective optimization problem. Its fitness function is the weighted summation value of multiple target components. By evaluating the path security and the path length respectively, the specific design of the fitness function is described as follows.

Combining the concept of the evaluation function with the behavior of the industrial robot described above, in the robot path planning, in order to reach the target point safely, or to reach the target point quickly, obstacle avoidance is an object that needs to be studied (Orozco-Rosas U et al.2015). Obstacles that need to be avoided are divided into static and dynamic, using the number N_{vs} of avoiding static obstacles and the number N_{vr} of avoiding dynamic obstacles, indicating the corresponding fitness function. The path security fitness function can be expressed as:

$$f_1 = k_1 N_{vs} + k_2 N_{vr} \quad (3)$$

In the formula above, k_1 and k_2 are weight coefficients. The shortest path is another important study condition for robot path planning. For a path with N nodes, D is the total length, which is the distance between the start and end points, then the fitness function is designed as Eq. (4).

$$f_2 = D = \sum \sqrt{(x_g - x_s)^2 + (y_g - y_s)^2} \quad (4)$$

This is the motion distance of the robot, which is the vector consisting of the starting point (x_s, y_s) , and the target point (x_g, y_g) . Combined with the two conditions mainly satisfied in the path planning, the above functions (3) and (4) are weighted and summed, and the comprehensive fitness of the system is G .

$$f = \omega_1 f_1 + \frac{\omega_2}{f_2} \quad (5)$$

In the formula, ω_1 and ω_2 are the weights of the path security degree, and they are used to represent the weights of the Q and W fitness functions, respectively. Since the length of the path is the primary factor determining the degree to which the robot reaches the target, the weight coefficients ω_1 and ω_2 satisfy $\omega_1 < \omega_2 < 1$.

IV. RESULT ANALYSIS AND DISCUSSION

In the research of mobile robots, the vehicle body often adopts a wheel structure, which has the advantages of easy modeling and good smoothing performance. The disadvantage is that the wheels may slip with the ground and affect the motion performance of the robot.

Since the current robot platform is only in the experimental stage, a three-wheel drive industrial robot was designed for testing, and the moving mechanism consists of independently driven three omnidirectional wheels. This system is unconstrained and can be moved linearly from one point to another and turned during travel to adjust the attitude angle. Its trajectory can be attributed to a polyline. The body of the car has a diameter of 17cm and the height is 15cm; the wheel diameter is 54mm; the wheel landing position is 7.5cm away from the center of the car body.

The whole is divided into three layers. The communication receiving unit is placed on the top layer; the middle layer is the main control board, including the CPU control power supply and the motor drive; the lower layer is the motor and the battery. Because the weight of the car is mainly concentrated on the motor and the power supply, the two parts were placed on the bottom layer helps to reduce the center of gravity of the car and make the movement more stable. Since the system is still in the test phase, there is no ball hitting device installed in front of the car body. It is subject to further improvement in the future.

In the Matlab7.1 environment, the dynamic obstacle avoidance path planning method based on genetic algorithm is simulated to calculate the shortest path of the robot. In the experiment, the number of motion steps is used to measure whether the robot motion path is the shortest. The motion step is set to 2. Three robots, R_1 , R_2 and R_3 , were arranged to calculate from the position of the center point of the robot as the coordinate point, from the starting point

$S_1(14,10)$, $S_2(16,14)$ and $S_3(14,18)$ to the target point $G_1(25,18)$, $G_2(27,14)$ and $G_3(25,10)$. Dynamic obstacles in the environment are designed to be square. First, the relationship between the maximum fitness value and the average fitness value

is given in the Matlab7.1 environment as shown in Fig. 1. In order to verify the validity and feasibility of the algorithm, the population size popsize=50 and the termination algebra is $T=60$.

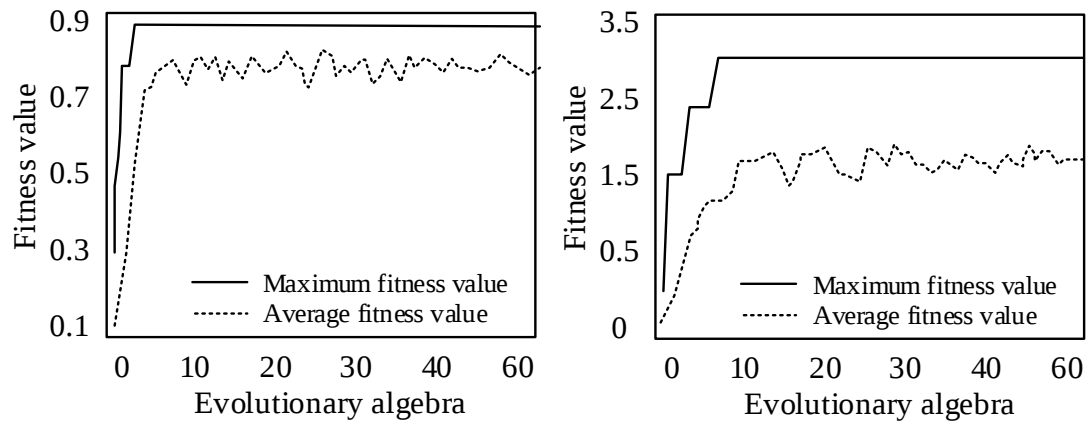


Figure 1. The evolutionary graph of maximum fitness function value and average fitness value.

The abscissa corresponds to the evolutionary algebra, and the ordinate corresponds to the fitness function. It can be seen that the average fitness value increases continuously until convergence. The selection probability is $P_r = 0.2$, the crossover probability is $P_c = 0.8$, and the mutation probability is $P_m = 0.01$. Taking $\omega_1 = \omega_2 = 0.5$, the population

evolved to stop after 60 generations, and the evolutionary convergence performance of the algorithm in the case of constant population size parameters was studied. The first experiment consisted of three robots. The convergence was shown in Table 1. The second experiment consisted of six robots. The convergence is shown in Table 2.

Table 1. The convergence of the 4th generation algorithm.

Algebra	4	8	10	30	60
Fitness value	0.88	0.88	0.88	0.88	0.88

Table 2. The convergence of the 9th generation algorithm

Algebra	4	8	10	30	60
Fitness value	1.78	2.51	3.05	3.05	3.05

It can be seen that the number of robots increases, the search space increases, and the convergence rate of the algorithm does not change much. It can be seen that the algorithm can also handle the case that the number of robots is large, that is, the number of obstacles increases. The behavior control parameters

of the optimized robot are obtained by genetic algorithm. The weighting coefficient values of each evaluation index in each robot fitness function are selected from the actual process. The control parameters of the three robot behaviors are shown in Table 3:

Table 3. Behavior control parameters.

Robot	λ_1	λ_2	λ_3	λ_4
R ₁	5.59	85.36	72.00	75.47
R ₂	28.39	33.18	20.18	52.58
R ₃	45.65	35.56	93.55	83.15

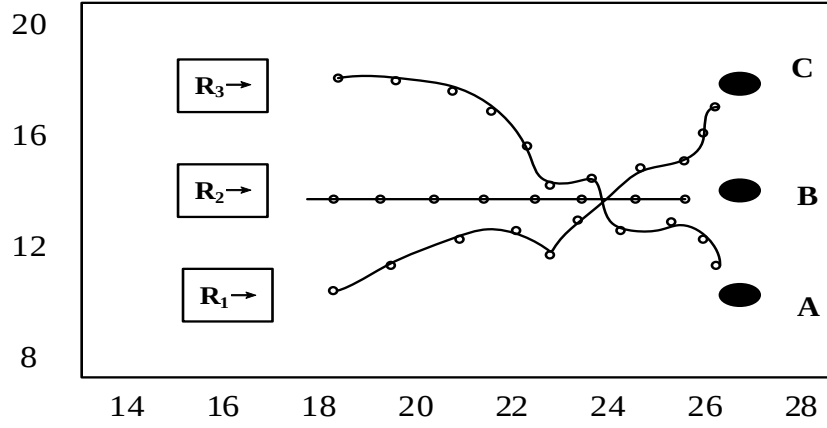


Figure 2. Simulation results.

Using the parameters shown in Table 3 for simulation, it can be seen that the three robots can advance toward the preset target and do not collide

with the static obstacle. R1's journey to the target point is 9 steps, R2's journey to the target point is 7 steps, and R3's journey to the target point is 8 steps

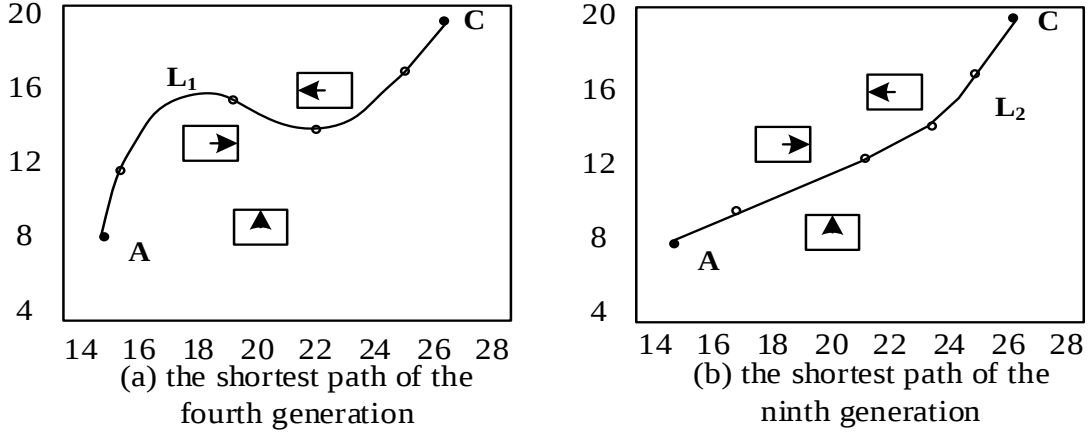


Figure 3. Shortest path.

For the experimental simulation of dynamic obstacle avoidance, it is assumed that the genetic algorithm parameters are set unchanged. With one robot as the research object, three starting points and three target points are set, and other robots are dynamic obstacles. As shown in Fig. 3: The genetic algorithm is used to plan the path of the robot. (a) shows the situation when the algorithm runs to the fourth generation, and (b) shows the situation when the algorithm runs to the ninth generation.

In Fig. 3a, it takes 36 steps for the industrial robot to move from point A to point c, and it will collide with dynamic obstacles. In Fig. 3b, the industrial robot only needs 20 steps to complete the same path, and it is a safe path that will not collide with dynamic obstacles. It is known from the experiment that the method proposed in this paper improves the running time of the robot and can plan a safe and smooth path. Matlab environment for simulation experiments, using Matlab included mathematical functions and toolbox functions, is the combination of Matlab and object-oriented VC++6.0. Experimental simulation

results show that the movement process of multi-robot system in the environment can realize the collision free motion, robots will be in accordance with the algorithm of planning the path of the running, and did not collide with other robots; It can reach the target point at the same time in a short time. This algorithm improves the speed of path planning and satisfies the real-time requirement.

The multi-objective optimization problem in mobile robot path planning is analyzed in this chapter. In order to reach the target point as soon as possible in an environment with static and dynamic obstacles, four objectives of industrial robots are analyzed. On this basis, using the behavior synthesis strategy of multiple mobile robots, a behavioral control path planning method based on genetic algorithm is obtained, and the behavior control parameters are optimized. Therefore, when using Matlab environment for simulation experiments, the speed of search is greatly improved, and satisfactory results are obtained. The results show that this method is feasible.

V. CONCLUSIONS

Multi-objective dynamic mobile machine is taken as the research object in this paper, and the algorithm of robot path planning is introduced. Firstly, the typical methods of robot path planning are introduced, and their advantages and disadvantages are compared. A method based on genetic algorithm for behavior control is proposed, and the tasks performed by the robot are described by behavior. The individual behaviors are then designed to represent the synthesis of behaviors by normalized summation. The genetic algorithm is used to optimize the control parameters of each behavior, and the genetic algorithm is designed. The fitness function is represented by the obstacle avoidance and the shortest path. The weighted summation method is used to represent the final fitness function. The satisfactory results are obtained through simulation experiments. This method improves the speed and real-time of path planning. It shows the superiority of the method in robot path planning in dynamic environment, and can find a smooth and safe shortest path.

REFERENCES

- Abu-Dakka, F.J., I.F. Assad, R.M. Alkhdour, et al., "Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots", *International Journal of Advanced Manufacturing Technology*, 89(1-4), 1-18 (2016).
- Kim, J.W. and S.K., "Fitness switching genetic algorithm for solving combinatorial optimization problems with rare feasible solutions", *Journal of Supercomputing*, 72(9),1-23 (2016).
- Liu, Y., Cong M, Liu D, et al. "Trajectory Planning for Porcine Abdomen Cutting Based on an Improved Genetic Algorithm and Machine Vision for Industrial Robot", *Jiqiren/robot*, 39(3), 377-384 (2017).
- Montiel, O., R. Sepúlveda and U. Orozco-Rosas, "Optimal Path Planning Generation for Mobile Robots using Parallel Evolutionary Artificial Potential Field", *Journal of Intelligent and Robotic Systems*, 79 (2) 237-257 (2015).
- Orozco-Rosas, U., O. Montiel and R. Sepúlveda, "Pseudo-bacterial Potential Field Based Path Planner for Autonomous Mobile Robot Navigation", *International Journal of Advanced Robotic Systems*, 12 (81), 1-14 (2015).
- Pereira, N., A.F. Ribeiro, G. Lopes, et al., "Path planning towards non-compulsory multiple targets using TWIN-RRT*", *Industrial Robot*, 43 (4), 370-379 (2016).
- Ran, Z., H.L. Dong and K.L. Hong, "Mobile Robot Navigation using Optimized Fuzzy Controller by Genetic Algorithm", *International Journal of Fuzzy Logic and Intelligent Systems*, 15(1), 12-19 (2015).
- Song, B., Z. Wang and L. Sheng, "A new genetic algorithm approach to smooth path planning for mobile robots", *Assembly Automation*, 36(2), 138-145 (2016).
- Tavares, P, P. Costa, J. Lima, et al., "Multiple Manipulators Path Planning using Double A*", *Industrial Robot*, 43(6):657-664 (2016).
- Wang, Y., Liu Q. "Robot path planning method based on rough set theory and a genetic algorithm". *Agro Food Industry Hi Tech*, 28(1), 1972-1976 (2017).
- Wang, X., Y. Shi, D. Ding, et al., "Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning", *Engineering Optimization*, 48(2), 299-316 (2016).
- Weerakoon, T., K. Ishii, and A.A.F. Nassiraei, "An Artificial Potential Field Based Mobile Robot Navigation Method To Prevent From Deadlock", *Journal of Artificial Intelligence & Soft Computing Research*, 5(3), 189-203 (2015).
- Yoon, H.S. and T.H. Park, "Motion planning of autonomous mobile robots by iterative dynamic programming", *Intelligent Service Robotics*, 8 (3), 165-174 (2015).
- Zhang, X., Y. Zhao, N. Deng et al., "Dynamic Path Planning Algorithm for a Mobile Robot Based on Visible Space and an Improved Genetic Algorithm", *International Journal of Advanced Robotic Systems*, 13(3), (2016).
- Zhao, L., Y. Zhang and X. Hu X, "A novel collision detection algorithm based on grids enveloping for industrial robot simulations", *China Mechanical Engineering*, 28(3), 316-321(2017).

Received: December 15th 2017

Accepted: June 30th 2018

Recommended by Guest Editor

Juan Luis García Guirao