Path planning of a Mobile Robot Using Real-coded Genetic Algorithm Based Simultaneous Exploration

Chih-Jer Lin, Yen-Lin Chen, Cheng-Hsin Liu, and Shen Kai Yu
National Taipei University of Technology, Taipei, Taiwan 10608
E-mail: ylchen@csie.ntut.edu.tw

Abstract — **In mobile robot researches, path planning and obstacle avoidance plays a very important role and has been a very challenging research topic. For path planning, it should produce continuous path from the starting point to the destination without colliding obstacles. Therefore, we proposes a genetic algorithm to search the path with the shortest path in Labview environment. The difficulty of the genetic algorithms applied to the mobile robot is how to reduce the complexity of the genetic operations, and how to avoid the region optimal solution and adaptation of environmental change. Many researchers studied genetic algorithms to determine the optimal solution such as path planning, but the past literatures mostly used binary coding for the gene encoding. If the path is longer or the number of obstacles is larger, the binary coding will be a lengthy string of series. This will result in the longer evolution of computing time. As a result, we propose a novel method which is a serial number encoded as a gene encoding to effectively reduce the evolution of computing time for the path planning applications.**

Keywords: Labview, real-coded genetic algorithm, path planning, mobile robot.

I. INTRODUCTION

With the rapid development of the robotics industry, the mobile robot and intelligent robot technology has already been very extensive and mature. The challenging field of robotics has an enormous potential to support the industries in different fields generally. A robot can perceive and manipulate the physical world by means of different devices, which make it interact and work under various working environments, such as planetary explorations, manufacturing in industry or as a player in a movie. We can see many automated intelligent robotic applications in our lives on many occasions, such as driverless system, automated guided vehicle, and the popular clean wheel robots. Especially, the market for service robots has an excellent chance and many successful commercial cleaning robots have been studied and developed some years ago. Meanwhile, the mobile robots can work for private purpose, not only for cleaning works. The common enabling technology for a mobile robot is autonomous navigation in an unknown environment. However, the handling with radioactivity in a nuclear power plant or some radiology demanded requirements, concerning the compliance of safety measures. Usually, the measurements are done by a member of staffs, who has to invest a lot of time for the safekeeping of a contamination-free building. The applications of robots could be extremely timesaving, safe and reduces physically demanding occupations.

In mobile robot researches, path planning and obstacle avoidance plays a very important role and has been a very challenging research topic. For path planning, it should produce continuous path from the starting point to the destination without colliding obstacles. However, many literatures have studied about obstacles-avoidance methods, such as the potential field method [1-4], certainties grid [5], and wall-following [6]. The above methods don’t study the shortest path. Therefore, we will use the genetic algorithm to search the path with the shortest path in Labview environment. The difficulty of the genetic algorithms applied to the mobile robot is how to reduce the complexity of the genetic operations, and how to avoid the region optimal solution and adaptation of environmental change. In [7-10], many researchers studied genetic algorithms to identify the optimal solution such as path planning, but the past literatures used binary coding for the gene encoding. In this way, if the path is longer or the number of obstacles is larger, the binary coding will be a lengthy string of series. This will result in the longer evolution of computing time. To overcome the above-mentioned drawbacks, this study proposes a novel method which is a serial number encoded as a gene encoding to reduce the evolution of computing time for the path planning applications of mobile robots.

II. HARDWARE and Ga METHODS

A. Hardware Architecture

In this paper, we use the robotino to implement the optimum path planning and that is developed by FESTO Company for training stage of mobile robot. Figure 1 shows the robotino that composes of three DC motor to move forward and backward and rotation. Robotino can feed back signal from sensor, such as analog sensors or digital sensors. The analog sensors include nine IR sensors and Inductive sensors, while the digital sensors include CCD and light sensors. In order to integrate other component, robotino provides eight digital inputs, eight digital outputs, eight analog inputs and two relays. And the controller is PC104 with Linux OS and real time kernel. The control of robotino divided into 2 types. One is the remote control (Fig.2), FESTO provides some LabView program to control Robotino, the program are calculated on the remote computer. The other one is that the program can be download into robotino to run by itself, and make a connection into the internal computer on the robotino. The firmware program is developed by the C programming language, and compiled for exectuing on the Linux OS.

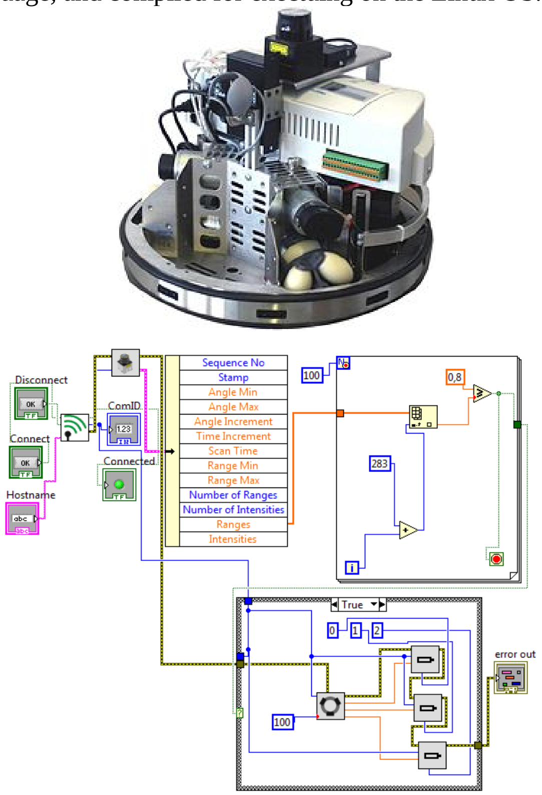


Figure 1 LabView remote control program.

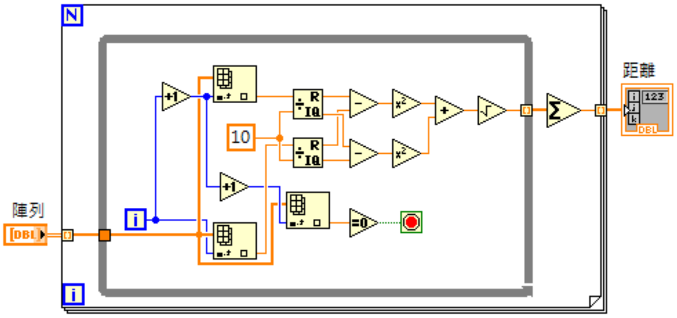


Figure 2 shows fitness function program

B. Initial Population

When program being running, the GA must randomly initialize the chromosomes that represent trajectories of the robot from the departure to the destination. Each chromosome need to include two conditions; the continuity and randomness of the trajectory should be optimally preserved. So the chromosome is designed to reveal the two features: 1). the chromosome length is unequal between each of the chromosomes; 2). A point is randomly chosen between the departure and the destination. Then the generated trajectory must pass this random point from the departure to the destination.

C. Fitness Function

In the genetic algorithms, fitness function will deice which chromosome is chosen into the next population. Fitness function effects performance of GA and calculation of time. In this paper, fitness function formula is following

$$f(P_i) = \sum_{j=0}^{Pop\_\mathrm{Chromosome}} \sqrt{(X_{j+1,i} - X_{j,i})^2 + (Y_{j+1,i} - Y_{j,i})^2} \quad (1)$$

,where p is the length of trajectory, $X_{j+1,i}$ is the position of the next X, $X_{j,i}$ is the position of the current X, $Y_{j+1,i}$ is the position of the next Y, $Y_{j,i}$ is position of current Y. So the summation of the total distance of gene and evaluates the optimum chromosome and re-searches the best one in the next population. Figure 2 shows the computation for the fitness function in Labview environment.

D. Competition Selection

The selection component applies the comparison of the fitness function value between two kinds of chromosome and then keep the best one. Generally, the compaction selection can choose the optimum fitness value that can into next stage, crossover or mutation. The selection step is given as follows:

Step 1: Calculate the fitness function of each chromosome by the formula (1).

Step 2: Randomly select any chromosome into crossover pool.

Step 3: compare fitness function of chromosome. If the fitness value of Pi less than Pj, then Pi will be selected. Otherwise, the Pj will be selected. Figure 3 shows the computation process of the competition selection for RGA in Labview environment.

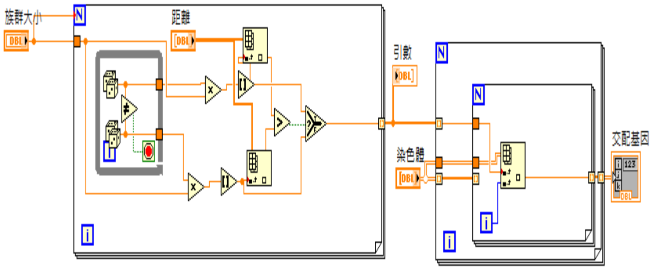


Figure 3 Competition selection program.

E. Crossover

The crossover component is to change the gene of two chromosomes and generate new one. The crossover component can have several methods, such as the single-point crossover, multi-point crossover or uniform crossover. In this paper, when the crossover rate is less than number by random, then the system will enter the crossover mode. Assume that the crossover mode is active, we have to search the same serial number from each chromosome because the trajectory must be continuous. When the serial number is same, we use single-point crossover to change two chromosomes. Figure 4 shows the computation for the crossover operation for RGA in Labview environment.

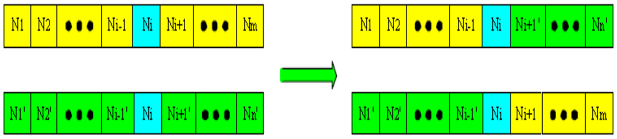


Figure 4 crossover operation (signal-point).

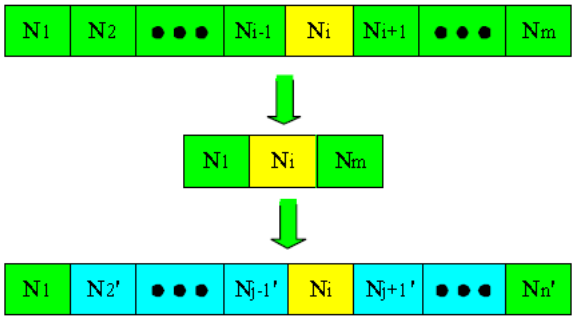


Figure 5 mutation operation

F. *Mutation*

Function of mutation component is add new gene to replace the old genet in the chromosome randomly. The solution of fitness that is it the best one isn't important. This paper presents a new mutation method that chooses a gene in chromosome and generate a new trajectory. When the mutation rate less than number by random then mutation mode is active. In the chromosome, we randomly choose a gene that means trajectory must pass this serial number. Figure 5 shows the computation for the mutation operation for RGA in Labview environment.

G. *Elitist Strategy*

The elitist strategy component is to keep the optimum solution after the selection, crossover and mutation processes that generate the new solution. The solution which include the parent and offspring will keep a half that is better than other. And then generate a half new one.

III. Results and Discussion

In order to simulate work environment of robot, the work space is will build by a 2D array, that describe the necessary information, such as obstacle sizes and positions. According to robotino dimension, each element size is defined that have to make sure that the robotino can easily past.

In the 2D workspace environment, the black box represent wall and obstacle and robot can't pass. The red point denotes departure of robot, and the blue point denotes destination. We calculate the moving trajectories of the robot by the proposed GA-based path planning approach, and archive them on the robotino.

Figure 6 denotes system flow chat that includes the proposed GA-based method and the motor commands. First, the2D simulate the work environment is built with obstacles. Next, the system waits until receive start command. Thirdly, the chromosome is initialized and the trajectory is generated. Afterwards, the fitness function is computed and then the crossover, mutation, elitist strategies are applied to keep the optimum solutions. As a result, the end condition can be checked and the robotino can move to the destination according to the determined path.

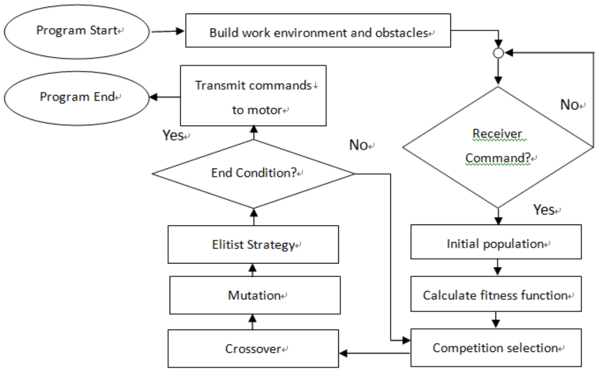
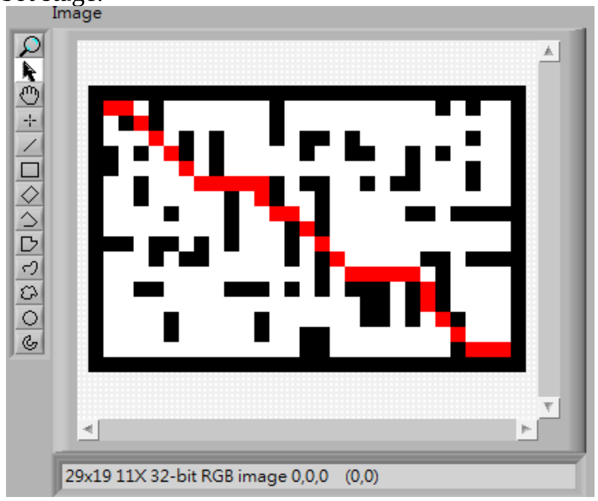


Figure 6 system flow chat

Figure 7 shows optimum path panning by GA. the departure of robotino is in the top left and the destination is at the bottom right. In this study, the population number of simulation is set as 1000, the crossover rate is set as 0.8, the mutation rate is set 0.02, and the end condition is to run 1000 populations. When GA finds the shortest path in current population, this chromosome will be recorded in memory and show in Fig. 8(a). Figure 8 (b) shows each population optimum fitness data in an overview. The simulation results demonstrate that the proposed GA-based path planning method can effectively achieve a shortest path, and we also verification the results on the FESTO robot stage.



29x19 11X 32-bit RGB image 0,0,0 (0,0)

Figure 7 Trajectory by GA panning

ACKNOWLEDGMENTS

This work is supported by the National Science Council under Grant NSC-100-2221-E-027-031, NSC-101-2218-E-033-003, 102-2219-E-027-006, and the Ministry of Economic Affairs, R.O.C. under Grant 98-EC-17-A-16-S1-127.

REFERENCES

[1]O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,”Proceedings of the IEEE International Conference Robotics and Automation, vol. 5,no. 1, pp. 90-98, 1986.

[2]J. Borenstein, and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,”IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, pp. 1179-1187, 1989.

[3]J. Barraquand, B. Langois, and J. C. Latombe, “Numerical potential field techniques for robot path planning,” IEEE Transactions on Systems, Man, and Cybernetics, vol.22, pp. 224-241, 1992.

[4]Y. Koren, and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” Proceedings of the IEEE International Conference Robotics and Automation, vol. 2, pp. 1398-1404, 1991.

[5]F. Vacher, “Fast local path planner in certainty grid,” Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2132-2137, 1994.

[6]P. Turennout, G. Honderd, and L. J. Schelven, “Wall-following control of a mobile robot,” Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 280-285, 1992.

[7]K. Sugihara and J. Smith, “Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot,” in Proceedings of IEEE international Symposium onComputational Intelligence in Robotics and Automation, pp. 138-143,1997.

[8]Ming-Yi Ju and Ching-Wen Cheng, "Smooth Path Planning Using Genetic Algorithms," Proceedings of the IEEE on the 8th World Congress on Intelligent Control and Automation, pp. 1103-1107, 2011.

[9]Hagen Burchardt and Ralf Salomon, "Implementation of Path Planning using Genetic Algorithms on Mobile Robots," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1831-1836, 2006.

[10]Kamran H. Sedighi, Kaveh Ashenayi, Theodore W. Manikas, Roger L. Wainwright and Heng-Ming Tai, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm," Proceedings of the IEEE Congress on Evolutionary Computation, vol.2, pp.1338-1345, 2004.

[11]I. Ashiru, C. Czarnecki, and T. Routen, “Characteristics of a Genetic Based Approach to Path Planning for Mobile Robots,” J. Network and Computer Applications, vol. 19,pp. 149-169, 1996.

[12] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, “Adaptive EvolutionaryPlanner/Navigator for Mobile Robots,” IEEE Transactions on EvolutionaryComputation, vol. 1, no. 1, pp.18-28, April 1997.

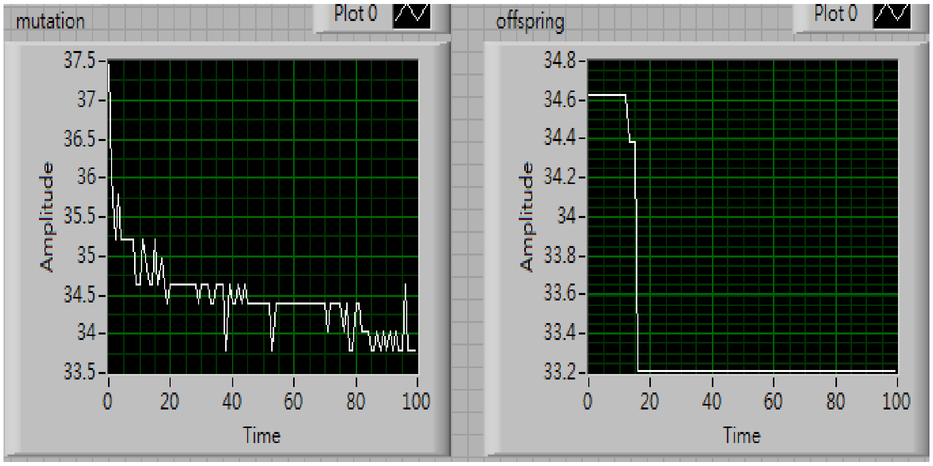


Figure 8(a) chromosome in RGA 8(b)The convergence of the RGA