Article ID: 1000 - 8152(2003)05 - 0657 - 07

Vector field based robot navigation using hybrid genetic/simulated annealing algorithm

ZOU Xi-yong[1,2], ZHU Jing[1]

(1. College of Electrical Engineering, Zhejiang University, Zhejiang Hangzhou 310027, China;
2. National Laboratory of Industrial Control Technology, Zhejiang University, Zhejiang Hangzhou 310027, China)

Abstract: An analytical vector field model of robot workspace was presented. In the model, the vector of resultant field described the most promising direction of robot motion. The model assumed that the edges of every obstacle, which was polygonal, were uniformly charged. It was shown that the resulting repulsive force, which pushing the robot away from the obstacles, could be calculated in closed form. Several factors including the length, the smoothness and the safety of the path require considering in robot navigation. Thus, a hybrid optimization algorithm, HGSA, which incorporated the simulated annealing algorithm (SA) into the genetic algorithm (GA), was proposed to optimize the path through searching the model parameters. The effectiveness of the proposed model was verified by computer simulation in three workspaces with different obstacle distribution. Comparisons between the optimized results show that the hybrid algorithm obtains better path solutions than either GA or SA.

Key words: vector field model; robot navigation; GA; SA

CLC number: TP301 **Document code**: A

基于混合遗传模拟退火算法的矢量场机器人导航

邹细勇[1,2], 诸 静[1]

(1.浙江大学 电气工程学院, 浙江 杭州,310027; 2.浙江大学 工业控制技术国家重点实验室, 浙江 杭州 310027)

摘要: 提出了一种解析形式的机器人矢量场导航模型,模型中场矢量的指向就是机器人的理想移动方向.模型假设工作空间中的障碍物为多边形,通过对障碍边界上电场的积分得到了排斥场的封闭解.导航必须考虑路径对长度、平滑度及安全性的要求,因此,一种混合遗传模拟退火优化算法被用来对导航模型的参数进行搜索,以寻找最优路径解.仿真结果验证了本文模型的有效性,优化所获路径的比较说明此混合算法要优于遗传算法和模拟退火算法.

关键词: 矢量场模型; 机器人导航; 遗传算法; 模拟退火算法

1 Introduction

Autonomous navigation is one of the most important topics in the robot area and can be categorized into two parts: reactive navigation[1,2] and path planning, the first one being local path planning based and the second one plans a path in the global workspace.

Local path planning is an on-line obstacles avoidance strateyy using the environmental information from its perceptual system and does not need a prior model of the environment. But the global path planning generates the overall path with much prior information on the environment. A complete path is required to be planned from its initial location to the goal position while avoiding collisions with obstacles. Commonly the first step in the planning process is to map the workspace into the configuration space (C-space). In configuration space, the robot is represented as a point, and the location and orientation of the obstacles are known.

In the global path planning area, the graph searching and potential field methods are the most popular approaches used. The graph searching method[3,4] firstly sets up a graph showing free space and forbidden space where there are obstacles. Based on this graph, a path is then selected by piecing together the grids or cells in the

Received date: 2002 - 12 - 17; Revised date: 2003 - 06 - 11.
Foundation item: supported by the National Natural Science Foundation of China (69904009).

free space.

A variety of potential field methods[5~7] have been presented in the past few years, positive (repulsive) potential fields are placed around the obstacles, and negative (attractive) potential field is placed at the goal position. The resulting field of these two ones acts on the robot to determine the motion direction. In general, the potential is a scalar function of the two distances, which are from the boundary of the obstacles to the robot and from the goal to the robot, respectively. The gradient of such a scalar function can be used as a fictitious force to navigate the robot.

In this paper, a new vector field model for robot navigation is presented, which in fact is an improved potential field model. In the workspace with polygonal obstacles, integral calculation of the repulsive field has been derived in closed form, thus the resulting repulsive field is constructed analytically, rather than numerically.

In order to take into consideration various requirements in robot navigation, a hybrid Genetic/Simulated Annealing algorithm is proposed to search for the preferred model parameters.

2 Vector field model

In the vector field model, the robot is denoted by a positive point charge, and negative charges are uniformly distributed on the edges of obstacle, while a negative point charge is placed at the goal position.

2.1 Attractive field of goal

The attractive field contributes to global navigation for the robot. If the field strength is set to be proportional to the distance between the robot and the goal, the motion may lose the direction in the area far apart from the goal. Also, if it is set to be inversely proportional to the distance, when the obstacles that are near the goal have strong field strength, the robot cannot approach the goal for the stronger repulsive force. So it is set to be a constant:

$$E_G(r) = k_1 \cdot \frac{Q_G}{|r_G - r|} \cdot (r_G - r), \qquad (1)$$

where r and r_G are the current position of the robot and the goal, respectively. Q_G is charge quantity of the goal, k_1 is a coefficient of the field strength.

2.2 Repulsive field of obstacles

The repulsive field contributes to local navigation for the robot. In traditional repulsive field model, the field strength of the point charge trends to be the value of ∞ when the point approaches the charge. But the robot may also collide with the obstacles if the motion speed is too high. An improved field model is developed (as shown in Fig. 1), the region where the field strength is infinite is some distance (D_1) shifted outwards. Thus the safety of the robot trajectory becomes tunable through changing D_1. The repulsive field of a point charge on the obstacle edge is defined as follows:

$$E_o(r) = \begin{cases} \dfrac{k_2 \cdot Q_o}{(|r - r_i|^2 - D_1^2)^{\frac{3}{2}}} \cdot (r - r_i), & (D_1 < d(r) \leqslant D_2), \\ 0, & (d(r) \leqslant D_1,\ d(r) > D_2), \end{cases} \qquad (2)$$

where r_i is the location of a point charge on an edge of the obstacle, Q_O is its charge quantity, k_2 is a coefficient, D_2 is the function range of the field, $d(r)$ is the least distance between the robot and the edges of this obstacle, D_1 is a shift parameter, which is called the basic safety distance.

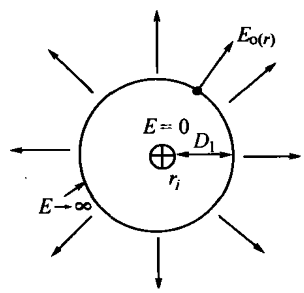

Fig. 1 Improved repulsive field of point-charge

It is assumed in the model that the obstacles are polygonal. An analytical solution of the field is obtained, through performing integration on point charge field along the obstacle edge. Considering an obstacle edge uniformly charged (as shown in Fig. 2), the line $\overline{AB}$ is the j-th edge of the i-th obstacle, it is denoted by $y = k_e \cdot x + d$. The solution of the repulsive field (in the function range) is:

$$E_{\overline{AB}}(r) = \int_{\overline{AB}} k_2 \cdot \frac{[(x_r - x) + j(y_r - y)] \cdot l_i}{[(x_r - x)^2 + (y_r - y)^2 - D_1^2]^{\frac{3}{2}}} \cdot \mathrm{d}s = E_{\overline{ABx}} + j \cdot E_{\overline{ABy}}. \qquad (3)$$

Where l_i is the line density of the charges on the obstacle edge and it is assumed to be 1.

$$E_{\overline{AB}}(r) = E_{ij}(r) = E_{ijx}(r) + j \cdot E_{ijy}(r) = \int_a^b k_2 \cdot \frac{[(x_r - x) + j(y_r - y)] \cdot l_i}{[(x_r - x)^2 + (y_r - y)^2 - D_1^2]^{\frac{3}{2}}} \cdot \sqrt{1 + k_e^2} \cdot \mathrm{d}x. \tag{4}$$

In this formula, the denominator is

$$(1 + k_e^2) \cdot \left\{ \left(x + \frac{k_e \cdot d - x_r - k_e \cdot y_r}{1 + k_e^2} \right)^2 + \left[\frac{(x_r^2 + y_r^2 + d^2 - 2y_r \cdot d - D_1^2)}{1 + k_e^2} - \left(\frac{k_e \cdot d - x_r - k_e \cdot y_r}{1 + k_e^2} \right)^2 \right] \right\}.$$

Let it be $= (1 + k_e^2) \cdot [(x + h)^2 + c]$. (5)

And define $k = \frac{k_2 \cdot l_i \cdot \sqrt{1 + k_e^2}}{(1 + k_e^2)} = \frac{k_2 \cdot l_i}{\sqrt{1 + k_e^2}}$, then

$$E_{\overline{ABx}} = k \cdot \left\{ \int_a^b \frac{x_r}{[(x+h)^2 + c]^{\frac{3}{2}}} \cdot \mathrm{d}x - \int_a^b \frac{x}{[(x+h)^2 + c]^{\frac{3}{2}}} \cdot \mathrm{d}x \right\} = \frac{k}{[(x+h)^2 + c]^{\frac{1}{2}}} \Big|_a^b \cdot \left\{ \frac{x_r \cdot (x + h)}{c} \Big|_a^b - 1 - \frac{h \cdot (x+h)}{c} \Big|_a^b \right\}, \tag{6}$$

$$E_{\overline{ABy}} = k \cdot \left\{ \int_a^b \frac{y_r}{[(x+h)^2 + c]^{\frac{3}{2}}} \cdot \mathrm{d}x - \int_a^b \frac{y}{[(x+h)^2 + c]^{\frac{3}{2}}} \cdot \mathrm{d}x \right\} = \frac{k}{[(x+h)^2 + c]^{\frac{1}{2}}} \Big|_a^b \cdot \left\{ \frac{(y_r - d) \cdot (x+h)}{c} \Big|_a^b + k_e + \frac{k_e \cdot h \cdot (x+h)}{c} \Big|_a^b \right\}. \tag{7}$$

Assume there are N polygonal obstacles in the workspace, and each obstacle has M_i edges, so the resulting field

$$E(r) = E_G(r) = \sum_{i=1}^{N} \sum_{j=1}^{M_i} E_{ij}(r) = E_{Gx}(r) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} E_{ijx}(r) + j \cdot [E_{Gy}(r) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} E_{ijy}(r)]. \tag{8}$$

The angle between $E(r)$ and axis X is defined to be α, then

$$\alpha = \mathrm{tg}^{-1} \left(\frac{E_{Gy}(r) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} E_{ijy}(r)}{E_{Gx}(r) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} E_{ijx}(r)} \right). \tag{9}$$

The quadrant of α is determined by the signal of $E_x(r)$. From formula (9), it is obvious that the computation complexity of this algorithm is linear with respect to the number of the obstacles edges within the environment, $O(NM)$, where M is the maximum edge number of a polygonal obstacle.

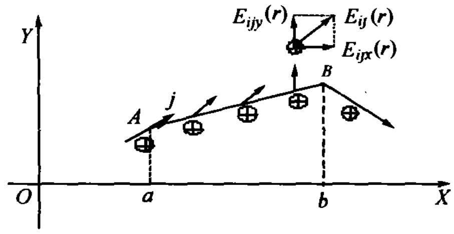

Fig. 2 Obstacle edge uniformly charged

3 Robot navigation with vector field model

3.1 Robot navigation method

Robot navigation is performed at a constant speed as follows:

Step 1 Generate the model parameters such as Q_G, k_1, k_2, D_1, D_2 and motion step-size S_w. Set up the field model for robot navigation;

Step 2 $k = 0$, set the present position to be the initial point $S(x_s, y_s): x_0 = x_s, y_0 = y_s$;

Step 3 Calculate the direction angle α according to formula (9), then the next position is

$$\begin{cases} x_{k+1} = x_k + Sw \cdot \cos \alpha, \\ y_{k+1} = y_k + Sw \cdot \sin \alpha. \end{cases} \tag{10}$$

Step 4 Steer the robot to the next position (x_{k+1}, y_{k+1}), reset the present position: $x_k = x_{k+1}, y_k = y_{k+1}$, $k = k + 1$;

Step 5 Judge if the robot reaches the goal position: $\sqrt{(x_k - x_G)^2 + (y_k - y_G)^2} < \mathrm{Tol}$ (Tol is a tolerable position error), if the condition is not satisfied and k is less than a preset biggest iteration number, return to Step 3;

Step 6 The robot reaches the goal, or the step of motion exceeds the preset biggest number, i.e., complete path cannot be found using this poor model.

As expected, the robot stops moving after reaching the goal position in the navigation. But if the parameters of the model are set inappropriately, a complete path cannot be found, sometimes even collisions will occur (see Fig.3). To the collision free paths, some other requirements must be taken into consideration in robot navigation, including the length, smoothness and the minimum & average distance from the obstacles of the path.

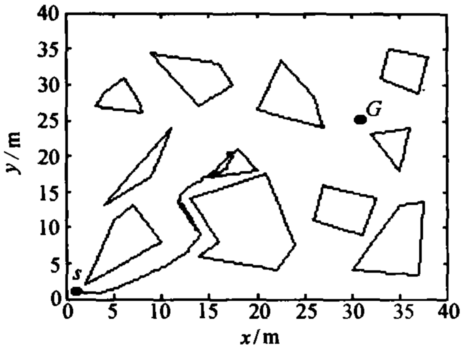

Fig. 3 Planned path with poor parameters

3.2 Collision detection for path

To detect whether there are collisions with obstacle on the path, here consider a normal instance as shown in Fig.4. In the figure, the line $\overline{AB}$ is an obstacle edge, and the curves marked with small hollow diamonds are two possible robot path. Points p_1 and p_2 are the nearest points to points A and B on axis X, respectively. It can be seen that the possible collision area is $[p_1, p_{2+1}]$ on the path for this edge.

First of all, find these two points p_1 and p_2 on the path. Assume that the line $\overline{AB}$ is $y = a_1 \cdot x + b_1$, and for every point k in this area $k \in [p_1, p_2]$, the line from point k to point $k+1$ is $y = a_2 \cdot x + b_2$. If $a_1 = a_2$, the two lines are parallel, no collision may occur; else calculate the cross point of these two lines:

$$(x, y) =$$
$$((b_2 - b_1)/(a_1 - a_2), a_1 \cdot (b_2 - b_1)/(a_1 - a_2) + b_1).$$

Lastly, judge ($x_A < = x < = x_B$ & $x_k < = x < = x_{k+1}$), if the condition is satisfied, there has a collision between this path section and the edge $\overline{AB}$, else no collision occurs.

For every obstacle edge in the workspace, repeat the process. Once a collision is found, stop this procedure, and there is no need of more detection for the rest edges.

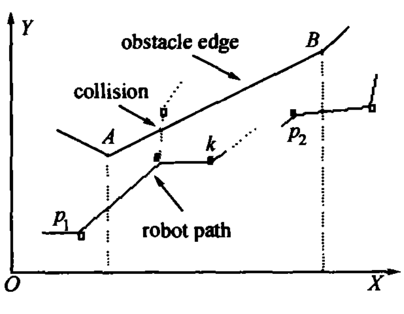

Fig. 4 Collision detection for robot path

4 Hybrid genetic/annealing algorithm for robot navigation

4.1 Hybrid optimization strategy

Since the robot navigation is an optimization problem (OP), some optimization techniques are used to search for optimal model parameters and motion step-size, including GA and SA.

GA[8,9] and SA[10] are two useful stochastic techniques, capable of solving the optimization problem approximately. GA is based on natural genetics and natural selection, and it is naturally parallel. Generally, a GA has three operators, starting from several to many points by reproduction, crossover and mutation, better solutions can be found rapidly with respect to the original population. Simple as it is, it suffers from poor convergence and usually has the inferior solution quality compared to the SA. SA is an optimization technique, which simulates the physical annealing process of a molten particle starting from a high temperature. It has the ability to escape local minimum by incorporating the probabilistic acceptance technique, but it usually takes much computation time in order to arrive at a near-global minimum and cannot easily exploit parallelism.

In this paper, a hybrid optimization algorithm (see Procedure 1), combining both GA and SA, is presented. The new algorithm, referred to as hybrid Genetic/Simulated Annealing algorithm (HGSA), trying to combine local and global searches, adopt the probabilistic acceptance of SA in the schedule to improve the convergence of the simple GA.

Procedure 1

The hybrid Genetic/Simulated Annealing Algorithm (HGSA)

Step 1 Initialize the parameters such as the population size, the crossover rate, the mutation rate and the cooling rate (η). Set initial temperature T_0. And generation number $k = 0$;

Step 2 Determine the value interval for every parameter in the field model and how they are coded into seven sub-strings, which are used to form the chromosome. Randomly generate the initial population $P(0)$;

Step 3 $k = k + 1$, for each individual in $P(k-1)$, decode the chromosome to the parameters of the field model, which then is used to navigate the robot.

Calculate the value of the fitness function according to the planned path (refer to Section 4.2);

Step 4 Implement selection, crossover and mutation operation on $P(k-1)$ to create $P'(k)$, and update the best solution found so far if possible;

Step 5 For each individual j in $P'(k)$, $i=0$,

While $i < S$

{

Generate a neighbor solution j' from j randomly, calculate the fitness value of j', replace j with j' with the probability $\min\left\{1, \exp\left[\frac{-(f_{j'}-f_j)}{T_{k-1}}\right]\right\}$, update the best solution found so far if possible;

$i = i + 1$

}

End for

S is a preset number for the SA loop. The population after this operation is the next generation $P(k)$;

Step 6 Decrease temperature $T_k = T^*_{k-1}\eta$;

Step 7 If the stop criterion has not been satisfied, return to Step 3;

Step 8 Record the best solution. Plot the path, which is planned with the parameters decoded from this solution.

4.2 Fitness value calculation

Considering the global aim is to plan a collision free path of minimum length, maximum smoothness, and maximum safety, four fitness sub-functions have been defined:

- Function of the length: $f_1 = \text{step}s \cdot Sw$;
- Function of the smoothness: $f_2 = rd\_\text{back}$;
- Function of the minimum distance to obstacles: $f_3 = 1/d_{\min}$;
- Function of the average distance to obstacles: $f_4 = 1/d_{\text{ave}} = 1/D_1$;

Where steps is an index of motion steps for the path, it is set as follows:

$$\text{steps} = \begin{cases} s\_\text{steps} & (s\_\text{steps} \leqslant 200 \ \& \text{ collision free}), \\ 200 + s\_\text{steps} & (s\_\text{steps} \leqslant 200 \ \& \text{ collision occur}), \\ 500 & (s\_\text{steps} > 200, \text{i.e., no finished path}), \end{cases} \quad (11)$$

where 200 is set for the 40×40 workspace, $s_$steps is the iterative number in the navigation. And $rd_$turn, a record of the coarseness on the path, is the number of the points, where the turning angle exceeds $\pi/6$ in two sequential steps in the path, or the two sequential turning angles are not in the same direction (clockwise or counter clockwise). $d_{\min}$ and d_{ave} are the minimum and average distance between the path and obstacles, respectively, d_{ave} can be denoted by D_1.

Linearly combine the four sub-functions into a composite scalar fitness function as the following weighted sum approach (w_i can be set according to the workspace and the preference of decision maker):

$$\begin{aligned} \text{fitness}(k) &= \\ & w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3 + w_4 \cdot f_4 = \\ & w_1 \cdot \text{steps} \cdot Sw + w_2 \cdot rd\_\text{back} + \frac{w_3}{d_{\min}} + \frac{w_4}{D_1}. \end{aligned} \quad (12)$$

So the optimization objective is to search for the least value of the fitness function.

5 Computational results & discussion

A series of computer simulations are conducted to evaluate the ability of the vector field model and HGSA to plan a path between two desired locations in three different workspaces. The computation is implemented on a Pentium computer and the program is coded in M file using Matlab.

The performances of the GA, SA and HGSA are tested with the same parameters and stop criteria, which the best solution found so far stays fixed at some consecutive generations. For the three optimization algorithm, the population size is set to be 50, p_c to be 0.8, p_m to be 0.02, and η to be 0.9. Each instance is randomly run 15 times for each algorithm.

Table 1 shows that the results obtained by HGSA are better than those obtained by SA and GA applied alone. The best solutions found so far of the HGSA have less fitness value than that of the other two; moreover, the CPU time of HGSA is much less than GA. It must be stated that the CPU time of the SA is less than the other two in our experiments. It will consume more time than the other two, if all the three algorithms are implemented serially[11].

It is difficult for simple GA to maintain a high diversity over time. As a result of the roulette wheel process, some best solutions duplicate themselves increasingly in the new generation, and low-fitness solutions gradually drop out. HGSA, by contrast, maintains a healthy diversity by using SA schedule, which accepts new solutions at greater rate with higher temperature, and protects good solutions from dropping out with lower temperature. It has been shown to overcome the poor convergence property of GA significantly.

With GA doing global search, HGSA can obtain a much larger portion of the solution space than SA. Thus it can arrive at better solutions in less time.

Fig. 5 plots the paths derived in three different workspaces from the start location (S) to goal position (G) using the model, whose parameters are decoded from the best solutions shown in Table 1. It can be seen that the paths are safe, short and smooth.

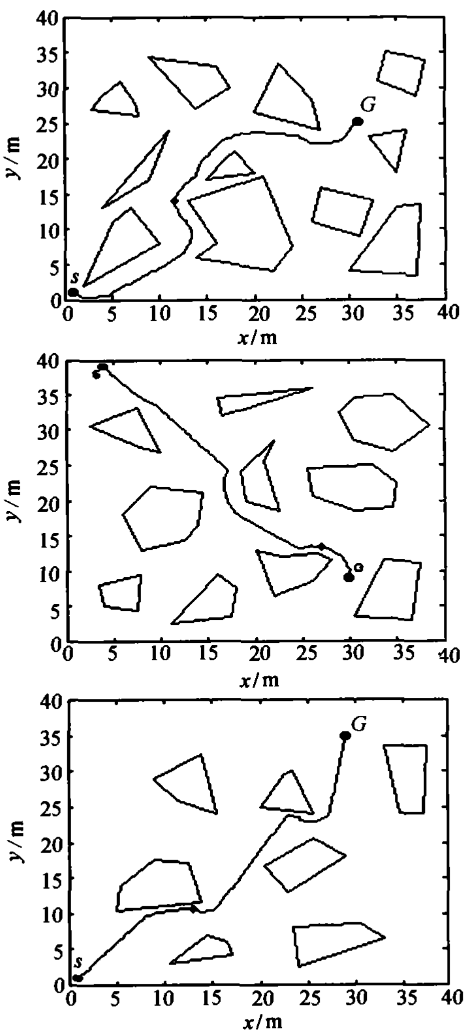

Fig. 5 Planned paths with the best model parameters found so far of HGSA in three environments with different obstacle distribution
(◆ denotes the nearest point to the obstacles)

Table 1 Comparison of results between GA, SA and HGSA

	GA		SA		HGSA	
	f^*	t_{ave}	f^*	t_{ave}	f^*	t_{ave}
Env.1	291.9517	4.4501	290.0510	1.7942	287.3537	2.5092
Env.2	264.7976	4.7913	263.7880	1.9054	261.3471	3.1549
Env.3	265.5475	3.4564	263.5825	1.4285	262.6091	2.6845

f^* : the best fitness value of the algorithm found over 15 runs;
t_{ave}: the average CPU time to obtain the optimal solution (in minute)

6 Conclusions

A novel vector field model is developed from an improved potential field for robot navigation in 2D workspace. An analytically rather than numerical solution of the repulsive field, has been obtained through performing line integration of point-charge-field along the obstacle edge. Besides collision, some other requirements must be taken into consideration in robot navigation, including the length, the smoothness and the safety of the path. In view that robot navigation is an optimization problem, a hybrid optimization technique, HGSA, is used to optimize the navigation model. Path planning tests in three workspaces using GA, SA, and the hybrid algorithm are presented. Simulation results show that the planned paths are very satisfying, using the proposed model. And it can be seen that the presented hybrid algorithm outperforms GA and SA.

For the prevalent drawback of local minimum in potential field based navigation algorithm, certain heuristic techniques are required to be appended in some unfavorable cases. The mending methods will be presented in the future work.

References:

[1] BALCH T, ARKIN R. Avoiding the past: a simple but effective strategy for reactive navigation [A]. *Proc of the Int Conf on Robotics and Automation* [C]. Los Alamitos: IEEE Computer Society Press, 1993, 1:678 – 685.

[2] BENREGUIEG M, MAAREF H, BARRET C. Fusion of fuzzy agents for the reactive navigation of a mobile robot [A]. *Proc of the Int Conf on Intelligent Robots and Systems* [C]. [s.l.]: [s.n.], 1997, 1:388 – 394.

[3] PODSEDKOWSKI L, NOWAKOWSKI J, IDZIKOWSKI M, et al. Modified A * algorithm suitable for online car-like mobile robot control [A]. *Proc of the First Workshop on Robot Motion and Control*

[C]. [s.l.]:[s.n.],1999:235-240.

[4] CHEN P C, HWANG Y K. SANDROS: a dynamic graph search algorithm for motion planning [J]. *IEEE Trans on Robotics and Automation*, 1998,14(3):390-403.

[5] VALAVANIS K P, HEBERT T, KOLLURU R, et al. Mobile robot navigation in 2-D dynamic workspace using an electrostatic potential field [J]. *IEEE Trans on Systems, Man and Cybernetics*, Part A, 2000,3(2):187-196.

[6] CHUANG J H. Potential-based modeling of three-dimensional workspace for obstacle avoidance [J]. *IEEE Trans on Robotics and Automation*, 1998, 14(5):778-785.

[7] GE S S, CUI Y J. New potential functions for mobile robot path planning [J]. *IEEE Trans on Robotics and Automation*, 2000, 16 (5):615-620.

[8] JAWAHAR N, ARAVINDAN P, PONNAMBALAM S G, et al. A genetic algorithm-based scheduler for setup-constrained FMC [J]. *Computers in Industry*, 1998,35(3):291-310.

[9] YAMAMOTO H. Robot path planning in FA by the scrap and build fitness method [A]. *Proc of Int Workshop on Robot and Human Communication* [C].[s.l.][s.n.], 1997:242-247.

[10] KATAYAMA K, NARIHISA H. Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem [J]. *European J of Operational Research*, 2001,134 (1):103-119.

[11] WONG K P, WONG Y W. Development of parallel hybrid optimization techniques based on genetic algorithms and simulated annealing [A]. *Proc of AI'94 Workshop on Evolutionary Computation* [C]. [s.l.]:[s.n.],1994:132-62.

作者简介:

邹细勇 (1979—),男,浙江大学电气工程学院博士研究生,主要研究方向为机器人控制,进化算法和计算机集成控制系统等,E-mail: zouxiyong@163.net;

诸　静 (1938—),男,浙江大学电气工程学院教授,博士生导师,主要研究方向为复杂系统智能控制,模糊控制和机器人控制等,获省、部级科技进步奖4项,出版教材与专著4部,发表学术论文80多篇,E-mail: zhujinghz@yahoo.com.cn.

(Continued from page 656)

mation Bachelor's degree from Huazhong University of Science and Techanology. Received Control Theory and Engineering Master's degree from Huazhong University of Science and Technology in 2000. Now she is a doctor candidate in Center for Nonlinear and Complex Systems in the department. Her research interests include fault diagnosis and fault-tolerant control of networked control system. E-mail: zyhidy@sohu.com;

FANG Hua-jing (1955—), male. He received the Ph.D. degree from Huazhong University of Science and Technoligy at the Departmeng of Control Science and Engineering in 1991 . Now he is a professor, doctoral advisor, the director of Control Theory Research Institute, and the vice-director of the Center for Nonlinear and Complex Systems. His current research interests include control system fault diagnosis, robust control and fault-tolerant control, analysis and synthesis of networked control system;

WANG Hua (1966—), male. He received the Ph.D. degree from University of Maryland at College Park in 1993. Now he is professor in Aeronautic and Mechanical Engineering Department, Boston University, U.S.A., and director in Center for Nonlinear and Complex Systems in HUST. His research interests include nonlinear control, intelligent control, networked control system, robot and sensor data fusion;

LI Li (1976—), male. He had graduated from Huazhong University of Science and Technology in 1999 with a B.S. degree. He is currently studying at Systems and Industrial Engineering Department, University of Arizona. His research fields include robust control and its application, intelligent controll and intelligent transportation.