Article ID: 1000 - 8152(2002)03 - 0363 - 06

Optimal Motion Planning Using Follow Boundary Repair and Evolutionary Search

YU Huiming[1], Chi Chia-Jung[1], QIANG Bi[1] and ZHOU Ganmin[2]

(1. Department of Computer Science, North Carolina A&T State University · Greensboro, NC 27411, USA;

2. Micro Electronic Research Institute, Hefei Polytechnic University · Hefei, 230009, P. R. China)

Abstract: This paper presents an optimal motion planning system for mobile robots operating in unstructured environments. We have developed a new obstacle representation method named cross-line, a follow boundary repair approach, and a hybrid evolutionary motion planning algorithm. A group of experiments are conducted that indicate the effectiveness of follow boundary repair approach and cross-line representation. These results also demonstrate that optimal/near optimal paths can be generated through combining the follow boundary repair and evolutionary search.

Key words: motion planning; follow boundary repair; hybrid evolutionary

Document code: A

采用跟踪边界修正和进化探索实现最佳的运动计划

于慧明[1] 季加荣[1] 秦 斌[1] 周干民[2]

(1. 北卡罗莱纳 A&T 州立大学计算机科学系·绿堡,美国; 2. 合肥工业大学微电子设计研究所·合肥,230009)

摘要: 研究在非结构化的环境中操作的移动机器人的最佳的运动计划系统,提出一种新的障碍表示法——交叉线,一种跟踪边界修正方法和一种混合进化运动计划算法.实验结果表明了跟踪边界修正方法和交叉线表示法的有效性,并且证明了结合跟踪边界修正和进化探索能够产生最佳的或接近最佳的路径.

关键词: 运动计划; 跟踪边界修正; 混合进化

1 Introduction

A mobile robot accomplishes given complicated tasks by moving in the real world. Motion planning, that is, deciding what motions to perform in order to achieve goal arrangements of physical objects, is one of the most important capabilities for a mobile robot. Although it may seem like a relatively simple job for humans, motion planning requires sophisticated integration of reasoning, perception and navigation. It has been an attractive research area in recent years.

Evolutionary computation (EC) is the field of research devoted to the study of problem solving via simulated evolution. Evolutionary search differs from other more traditional search in that it works with a population of candidate solutions (CSs). In most evolutionary computations, a population of CSs is randomly generated and allowed to evolve using a number of sexual and/or asexual procreation operators. The offspring of a generation usually replace weaker members of the population in order to keep the population size constant. EAs have been successfully applied to a variety of areas such as design optimization, machine learning, constraint satisfaction, and constrained optimization, to name a few[1]. Recently, there has been a growing interest in applying EAs to the area of motion planning[2~6].

The mobile robot motion planning problem can be described as that given a robot and a description of the environment, planning a path between the start and the destination which is collision-free and satisfies certain optimization criteria. The motion planning can be stated as follows: given an environment E(R, S, D, O) where R represents a point robot, S represents the start point (or current position of R), D represents the destination point and O represents a set of obstacles, find a collision

Received date: 2001 - 01 - 15; Revised date: 2001 - 12 - 29.

free path from S to D that R can traverse.

This paper will describe a new environment representation method named cross-line representation, a following boundary repair strategy and a hybrid evolutionary algorithm. Finally the implementation and experiment results will be presented.

2 Aimy – An autonomous mobile robot

Aimy is an autonomous mobile robot that consists of three main parts: a central control computer, a motor system, and a sensor processing system (shown in Fig.1). The central control computer is an IBM compatible 80386 notebook that is installed on Aimy's body. The communication between the central control computer and other systems takes place through different asynchronous I/O channels. Aimy's motor system uses a tricycle configuration consisting of three wheels, one at the front and two at the rear.

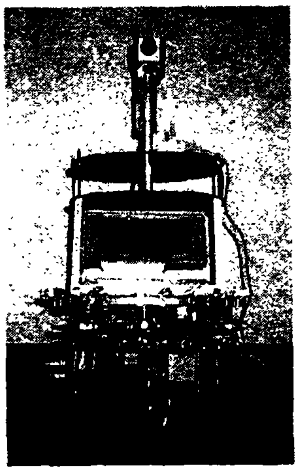


Fig. 1 Aimy—An autonomous mobile robot

3 Cross – line environment representation

Maps are necessary to support a mobile robot traveling in various environments. A map can be a free-space map, an object-oriented map, a composite-space map, a rule-based map, etc. In object-oriented maps obstacles are often represented as vertex graphs. In a vertex graph G(V, E), V is the set of all vertices of the obstacles in an environment, and E is the set of all edges connecting any two vertices in V (i.e. E is the set of boundary lines of obstacles). Fig. 2 shows examples of triangular, convex and concave obstacles. In order to support motion planning in continuous free space, vertex graphs are widely used to represent the obstacles of an environment.

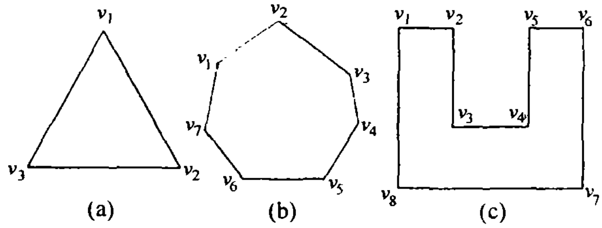


Fig. 2 Boundary line representation of obstacles

To decrease the complexity of calculation one may further decrease the computation complexity of path planning by using partial graph methods (PGMs). Based on previous work[2] we developed a cross-line method to represent different obstacles in various environments.

Cross-line representation classifies obstacles into three categories: triangular, convex and concave. The details are described as follows:

Category 1 If an obstacle is triangular it will be represented by its three boundary lines;

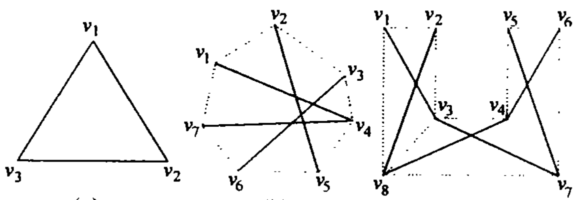
Category 2 If an obstacle is convex it will be represented as a set of $[T/2]$ line segments where T is the total number of vertices in the obstacle. Each line segment, $l_{i,k}$, connects two distinct vertices $e(v_i, v_k)$. Where i is the index of one vertex, k is the index of another vertex that equals to $(j + [T/2])$ where j is the index of next vertex of i.

Category 3 If an obstacle is concave the obstacle will be divided into several convex entities without adding any new vertices. These convex entities are represented using intersecting line segments as described in Category 2.

In cross-line representation, each vertex is assigned an angle value (AV). If the vertex connecting two edges forms an angle that is greater than 180 degrees the AV will be zero. If the angle is less than 180 degrees the AV will be 1. If an obstacle lies on the environment boundary the AV will be a large number 9999.

Figure 3 shows the three obstacles that are represented using the cross-line method. In Fig.3 (a), a triangle is described by lines $e(v_1, v_2)$, $e(v_2, v_3)$ and $e(v_3, v_1)$. In Fig.3 (b), a convex obstacle with seven vertices is represented using four cross lines that are $e(v_1, v_4)$, $e(v_2, v_5)$, $e(v_3, v_6)$ and $e(v_4, v_7)$. Fig.3 (c) shows an example representation of a concave obstacle. This obstacle is divided into three convex entities. Six cross lines are used. They are $e(v_1, v_3)$, $e(v_2, v_8)$, $e(v_3,$

v_7), $e(v_4, v_8)$, $e(v_5, v_7)$ and $e(v_6, v_4)$. The advantage of cross-line representation is that it reduces the total number line segments required to represent an obstacle. Therefore, it reduces the space and time complexity in discovering feasible paths.



(a) (b) (c)

Fig. 3 Cross-line representation of obstacles

4 Follow boundary repair algorithm

The follow boundary repair algorithm is based on human behavior. When a human walks on a street and encounters an obstacle he/she will quickly decide to turn either left or right. After turning, he/she will then follow the obstacle boundary, aim towards the original direction and make a short cut to continue the journey. Using this insight, a follow boundary repair heuristic was developed and used in the implementation of follow boundary repair.

During motion planning if an obstacle lies along a straight-line segment of a path between points P and Q, each line of the obstacle will be checked to see if it is intersected by the path segment P-Q. If a line of the obstacle is intersected with the path segment, follow boundary repair algorithm will be used to insert a new via point in the path. The follow boundary repair algorithm is described as follows.

Follow boundary repair algorithm

When P-Q has an intersection with the obstacle:

Case 1 If both end vertices with AV values zero, select the vertex that is closer to the intersection point.

Case 2 If the robot current standing point (robot position) is away from the obstacle, choose the vertex that is near the intersection point; if the vertex has AV greater than zero, find the closest vertex with an AV value of zero and select this vertex.

Case 3 If the robot current standing point has AV zero and direction is undecided, select the nearest vertex with an AV value of zero, set up direction.

Case 4 If the robot current standing point has an AV of zero and the direction is set, follow the direction to pick up the next point.

Repeat this process until the new line segment does not intersect this obstacle.

Each repair process is preceded by a checking procedure, which indicates whether a line P-Q intersects with an obstacle. If line P-Q intersects with an obstacle, then a new node (via point) R is generated. R is a point along an extension of a cross-line. The checking procedure then goes through lines P-R and R-Q. If any of them intersect with an obstacle, the repair process is repeated. Follow boundary repair algorithm is run recursively until there are no infeasible line segments along the path.

Figure 4 shows an example of how follow boundary repair repairs a concave obstacle. In Fig.4(a), P-Q line is intersected by the line $l_{4,8}$ and $l_{3,7}$. The point V_8 is selected based on Case 2. The result is shown in Fig.4 (b). The new P point is V_8. V_8-Q is intersected with line $l_{1,3}$, V_1 is chosen based on Case 3. The result is shown in Fig.4 (c). V_1-Q is intersected with line $l_{2,8}$, V_2 is selected based on Case 4. The result is shown in Fig.4 (d). Connecting V_2-Q the final path from P to Q is shown in Fig.4 (e) that is P - V_8 - V_1 - V_2 - Q.

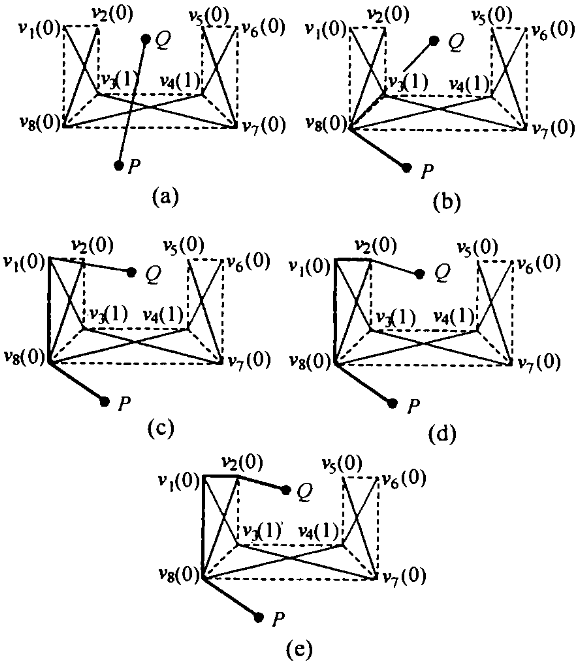


Fig. 4 Follow boundary repair for a concave obstacle

5 Hybrid evolutionary motion planning

A hybrid evolutionary motion planning algorithm has been developed that incorporates cross-line representation and follow boundary repair algorithm. Specific chromosome structures, genetic operators and fitness function have been designed and implemented. The details will be discussed in this section.

5.1 The hybrid evolutionary motion planning algorithm

In the hybrid evolutionary motion planning algorithm (HEMPA), a chromosome represents a path, consisting of straight-line segments, as a sequence of via points. A population consists of a number of (feasible and/or infeasible) paths leading from the start to the destination. Instead of relying on evolution exclusively, the HEMPA uses follow boundary repair algorithm to quickly transform infeasible paths into feasible ones. The HEMPA is described as follows:

begin

set iterate generation counter $g = 1$

input an environment with obstacles

initialize a group of paths Path (g)

use fitness function to evaluate each individual path in Path (g)

while ($g <$ total generation) **do**

begin

update the iterate generation counter g

select path (s) with low fitness value from Path (g)

produce a group of offspring by applying genetic operators to the selected paths

evaluate each new offspring

replace the worst member (s) of the population Path (g) by the produced offspring

use follow boundary repair approach to transfer r number of infeasible paths into feasible ones in Path (g)

evaluate new paths in Path (g)

end

select the best path from Path (g)

end

The HEMPA first reads an environment map that consists of obstacles, the starting location and the destination. Then an initial population of randomly generated paths is created. This population represents the first generation. Each chromosome is then evaluated. Tournament selection is used to select parents for procreation. With the proper number of parents selected, a number of offspring are produced by using a set of procreation operators. The offspring replaces the worst individuals in the population. Follow boundary repair is then used to repair the r individuals of the current population, where r is a user specified parameter that ranges from zero to the user-specified population size. The hybrid evolutionary process terminates after user-specified number of generations (iterations) and returns the best path discovered.

5.2 The fitness function

An evaluation function is used to assign a fitness to each candidate path (CP). The fitness of a CP is a measure of its "goodness". The evaluation function is as follows

$$\text{fitness}(P) = \alpha \cdot \text{dis}(P) + \beta \cdot \text{repair}(P),$$

$$\text{dis}(P) = \sum_{i=0}^{n-1} \text{segdis}(P_i, P_{i+1}),$$

$$\text{repair}(P) = \sum_{k=1}^{m} (l_{i,j})_k \times (m + n).$$

Total distance of a path is the sum of the Euclidean distance of each straight-line segment of a CP. For a feasible path the fitness value equals its total distance, while for an infeasible path the fitness value equals the sum of its total distance and the estimated follow boundary repair cost repair (P). The follow boundary repair cost is the sum of the products. Each product equals that the length of intersected obstacle representation line times a variable m or n. $l_{i,j}$ is the line that connects vertexes v_i and v_j. If the sum of AV_i and AV_j equals zero, n will be zero; otherwise m will be zero. In the HEMPA the lower the fitness value is, the better "fit" the individual.

5.3 The operators

Four evolutionary operators were designed. They are mutation, crossover, insertion and deletion. The application of each operator is probabilistic. Crossover combines two parent paths to create two offspring paths. It is applied on both feasible and infeasible paths. Mutation is used for fine tuning node coordinates in a CP for shape adjustment. It selects intermediate nodes in a path and changes the coordinates of this node randomly. When the selected path is feasible the change of the coordinates should be within the local clearance of the path so that the path subsequently remains feasible. When the selected path is infeasible a large random change should be imposed. Insertion operates on an infeasible path by inserting new nodes into infeasible path segments. The nodes are selected according to follow boundary repair. Deletion removes nodes from a path. If the path is infeasible, nodes for deletion are selected randomly in the path. Otherwise, the operator decides whether a node should be deleted based on the following ordered priority. 1) Line segments connected to the node are both infeasible. 2) One line segment connected to the node is infeasible. 3) If the selected infeasible path does not meet these conditions, the operation is aborted.

6 Experiments and results

The hybrid motion planning simulation system (HMPSS) has been implemented that consists of an interactive graphic user interface (GUI), environment, path data files and algorithms. The cross-line representation, follow boundary repair and hybrid evolutionary motion planning algorithm have been successfully simulated using HMPSS. A set of experiments has been conducted using a number of test environments with a variety of parameter settings. The results show that the motion plans evolved by the HMPSS allowed a mobile robot to reach the destination 100% of the time.

The HEMPA was run ten times on each of the test environments. A run was considered successful if the mobile robot reached the goal. In each test S represents the start point, G is the destination position. All obstacles are represented using cross-line representation. The test results are shown in Figs. 5 ~ 7. These figures display a group of snapshots of path evolution at different generations. For each snapshot only the ten best paths are displayed. For all tests the mutation, crossover, insertion and deletion operator usage rates were set to 0.25. The fraction of population to be replaced is 10 percent. Different complexities of the environment were reflected by the generation gap, population size, repair number and the number of generation.

Figure 5 shows the results of using evolutionary path planning without follow boundary repair, the number of generation is 10, generation gap of 0.1 and population size 10. Using these settings no feasible path was found. Next, we increased the population size to 500 and the number of generation to 20, kept generation gap the same, and disabled follow boundary repair. Fig.6 shows the feasible paths; however, they are neither optimal or near optimal. Fig. 7 shows the results of experiments where follow boundary repair was enabled, the generation gap was set to 0.1, population size was 10 and the number of generation was 10. The optimal/near optimal paths were evolved.

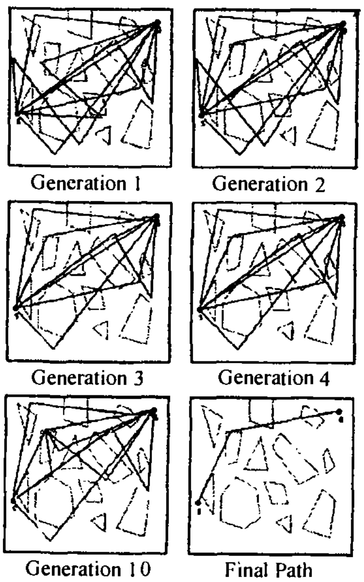


Fig. 5 Results without follow boundary repair (population size is 10, the number of generation is 10, generation gap is 0.1)

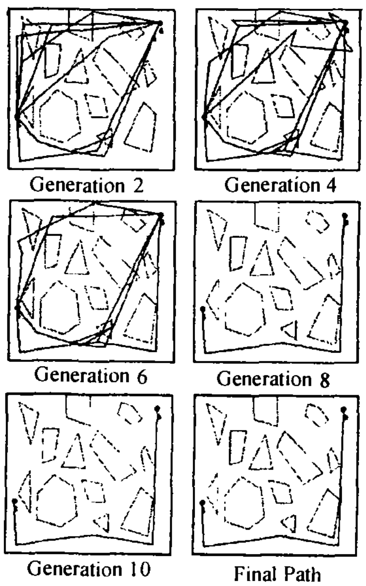


Fig. 6 Results without follow boundary repair (population size is 500, the number of generation is 20, generation gap is 0.1)

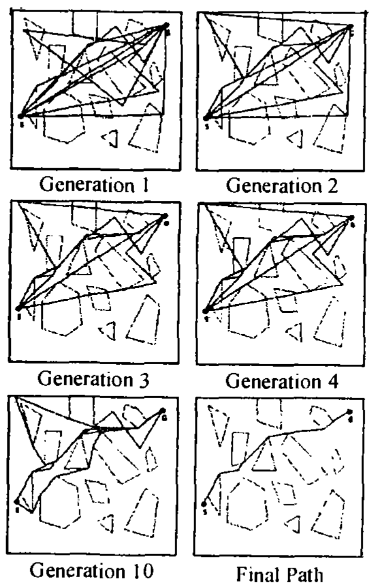


Fig. 7 Results without follow boundary repair (population size is 10, the number of generation is 10, generation gap is 0.1)

Table 1 shows a comparison of experimental results where follow boundary repair is either enabled or disabled, as well as the other parameter selections that include population size, the number of generation, generation gap and operator usage rates. For each environment the hybrid evolutionary algorithm was executed 20 times without follow boundary repair and with follow boundary repair receptively. In Table 1, the parameters were set up for a population size of 10, the number of generation 10, a generation gap 0.5 and operator usage rates of (crossover, mutation, insertion and deletion) 25%. For each environment the first column is the result without follow boundary repair, the second column is the result with follow boundary repair and the last column is the result of an optimal path. The first row is the average fitness value. The second row is the feasibility rate. The third row is the standard division. From the table we can see that without follow boundary repair the evolutionary algorithm can not guarantee to the finding of a feasible path, such as environment 2, 3, 4, and the feasible path rate is very low. The standard division is high. The hybrid evolutionary algorithm always finds a feasible path with a lower average fitness value.

Table 1 Comparison of motion planning with and without follow boundary repair

	Environment 1			Environment 2			Environment 3		
20 Trials Each	Repair(0)	Repair(1)	Optimal	Repair(0)	Repair(1)	Optimal	Repair(0)	Repair(1)	Optimal
Average	6800.02	5383.92	5139.22	13053.53	6966.72	6211.52	12701.17	5342.41	4455.50
Feasibility Rate	80%	100%		0%	100%		0%	100%	
Standard Deviation	1013.88	418.11		1674.88	616.99		2551.45	516.57	
	Environment 1			Environment 2			Environment 3		
20 Trials Each	Repair(0)	Repair(1)	Optimal	Repair(0)	Repair(1)	Optimal	Repair(0)	Repair(1)	Optimal
Average	26495.60	12038.30	12038.26	10014.25	5249.68	4756.69	7175.82	5416.08	5216.35
Feasibility Rate	0%	100%		5%	100%		75%	100%	
Standard Deviation	0	0		1448.96	417.84		845.89	357.19	

7 Conclusions

In this paper, we have used the cross-line method to represent various obstacles, follow boundary repair algorithm to quickly transform infeasible paths into feasible ones, and a hybrid evolutionary algorithm to implement mobile robot motion planning in various environments. We have designed and implemented hybrid evolutionary motion planning simulation system and conducted a number of experiments. The results show that follow boundary repair algorithm is very effective. The cross-line representation greatly reduces the memory requirement and calculation time of developing feasible path plans. The results show that the integration of cross-line representation, follow boundary repair, and evolutionary computation forms a powerful and efficient paradigm for motion planning and obstacle avoidance. These results also demonstrate correctness, effectiveness and efficiency of these method and algorithms. In the future we will apply these method and algorithms to the mobile robots.

References

[1] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs[M]. 2nd edition. USA: Springer-Verlag, 1994

[2] Ashiru I and Czarnecki C. Optimal motion planning for mobile robots using genetic algorithms [A]. Proceedings of IEEE /IAS International Conference on Industrial Automation and Control [C], Hyderabad, India, 1995, 297 - 300

[3] Dozier G, Esterline A, Homaifar A, et al. Hybrid evolutionary motion planning via visibility-based repair [A]. Proceedings of the 1997 IEEE Int. Conference on Evolutionary Computation [C], Indianapolis, IN, USA, 1997, 507 - 511

[4] Hocaoglu C and Sanderson A. Planning multi-paths using speciation in genetic algorithms [A]. Proceedings of the 1996 IEEE Int. Conference on Evolutionary Computation [C], Nagoya, Japan, 1996, 378 - 383

[5] Michalewicz Z. Evolutionary computation: Practical issues [A]. Proceedings of IEEE Int. Conference on Evolutionary Computation [C], Nagoya, Japan, 1996, 30 - 39

[6] Xiao J, Michalewicz Z, Zhang L, et al. Adaptive evolutionary planner/navigatror for mobile robots [A]. IEEE Trans. on Evolutionary Computation, 1997, 1(1): 18 - 28

(Continued on page 375)

On the basis of the technique of exact linearization, this paper has discussed the robust control of a class of SISO affine nonlinear systems with L_∞-bounded disturbances, and corresponding robust controllers have been obtained. In the final part of the paper, a simulation has been carried by MATLAB and SIMLINK, which illustrates the correctness of the results. Clearly, the results in the paper can further be generalized to the case of MIMO nonlinear systems, and the cases with more uncertainties, but further research is needed.

References

[1] Van Der Schaft A J. L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control [J]. IEEE Trans. Automatic Control, 1992,37(6):770-784

[2] Shen T L and Tamura K. Robust H_∞ control of uncertain nonlinear systems via state feedback [J]. IEEE Trans. Automatic Control, 1995,40(4):766-768

[3] Nquang S K. Robust H_∞-output feedback control [J]. IEEE Trans. Automatic Control, 1996,41(7):1003-1007

[4] Lu W M and Doyel J C. Robust analysis and synthesis of uncertain nonlinear systems [J]. IEEE Trans. Automatic Control, 1997, 42 (12):1662-1668

[5] Isdori A. Nonlinear Control Systems [M]. 2nd edition. Berlin: Springer-Verlag, 1989

[6] Wu M, Peng Z H, Tang Z H, et al. μ analysis and μ synthesis for nonlinear robust control systems [J]. J. of Central South University of Technology, 1998,5(1):68-73

[7] Peng Z H, Wu M and Cai Z X. I/O linearization based on μ synthesis for nonlinear robust control systems [A]. Proceeding of 14th IFAC Conference [C], Beijing, China, 1999,117-122

[8] Liu Y J and Qin H S. Global stabilization via dynamic output feedback for a class of uncertain nonlinear system [J]. Control Theory and Applications, 1999,16(6):830-835

[9] Hashimoto Y, Wu H S and Mizukami K. Robust output tracking of nonlinear systems with mismatched uncertainties [J]. Int. J. Control, 1999,72(5):411-417

本文作者简介

吴　敏　见本刊 2002 年第 2 期第 206 页.

张凌波　见本刊 2002 年第 2 期第 206 页.

刘国平　1962 年生.工学博士.1985 年研究生毕业于中南工业大学自动控制工程系,获工学硕士学位.1989 年赴英国曼彻斯特大学留学,1992 年获博士学位,并先后在英国约克大学、谢菲尔德大学和阿尔斯通电气公司工作,现为英国诺丁汉大学高级讲师.研究领域为:先进控制理论及应用,过程控制,鲁棒控制和网络控制等.

(Continued from page 368)

[7] Yu H and Malik R. Aimy: an autonomous mobile robot navigation in unknown environment with infrared detector system [J]. J. of Intelligent & Robotic Systems, 1995,14(2):18-28

本文作者简介

YU Huiming　Received her Ph.D. degree in computer science from the Stevens Institute of Technology, Hoboken, New Jersey, USA, in 1992. Currently she is an associate professor and also director of graduate studies in Department of Computer Science at North Carolina A&T State University. She was the principal investigator responsible for several robotics research projects. Her research interests include evolutionary algorithms, motion planning, robotics and artificial intelligence.

CHI Chia-Jung　Received his B.S. degree in physics from Chinese Culture University, Taipei, Taiwan in 1991 and M.S. degrees in human-machines system engineering and computer science from North Carolina A&T State University in 1997 and 1999, repectively. He is currently with Open Software Solutions, Inc., Greensboro, NC. His research interests include evolutionary algorithm in motion planning, human-computer interaction.

QIANG Bi　Received his master degree in computer science from North Carolina A&T State University, Greensboro, USA in 1999. From 1996 to 1997, he worked as an assistant engineer in Kezhi Co. Ltd., China. He is currently working for the Nationwide Insurance Company in the department of auto system.

ZHOU Ganmin　Received his master degree in microelectronics design from Hefei Polytechnic University, Hefei, China in 2001. He is currently studying for Ph. D degree in Microelectronic Research Institute of Hefei Polytechnic University. Now he is chiefly studying how to design the large-scale integratedcir cuit.　Email: gm_zhou@263.net