Mobile Robot Path Planning based on Probabilistic Model Checking under Uncertainties

Wei Lou[1, a], Chunrui Xia[2, b]

[1]School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

[2]Highly Reliable Embedded Systems Lab, Capital Normal University Beijing, 100048, China

[a]highouse1990@gmail.com, [b]xiachunrui@126.com

Keywords: Mobile robot; Path planning; Probabilistic model checking; Markov decision process; Probabilistic computation tree logic.

Abstract. In this paper, a probabilistic model checking method for mobile robots path planning problem is proposed. Since surroundings always affect the behavior of mobile robots, four main environmental factors are analyzed as influencing parameters. With the map built by randomized sampling-based method, we model the uncertain motion behavior as a Markov Decision Process (MDP). Meanwhile, the properties are described in PCTL (Probabilistic Computation Tree Logic) which can be used to describe rich mission specifications. Then the path planning problem is mapped to the problem of generating an MDP control policy that maximizes the probability of accomplishing the mission objective satisfied a PCTL formula. We apply the PRISM platform to analyze model and verify properties. Our approach is demonstrated with illustrative case studies.

1. Introduction

Path planning is an important research foundation of the basic application in robot control system, so it has always been a research focus in this field, and a lot of research achievements have been made. Path planning problem is to search an optimum motion path for a collision-free path that connects a given start and goal configuration while satisfying constraints imposed by complicated obstacles [1-2]. Most of the robot's working environment is dynamic and uncertain environment [3], so this paper is based on a static global path planning, we propose a dynamic obstacle collision avoidance strategy. Upon completion of the robot global path planning based on the information on static obstacles, dynamic collision avoidance prediction, and then the local collision avoidance planning.

There has been an increased interest in applying tools from automata theory and formal verification for planning and control of mobile robots, and have achieved significant achievements in practical application [3-4]. For example, in paper [5-6], the motion planning problem is formulated as a Constrained Markov Decision Process with mission specification to control a vehicle in a dynamic and threat-rich environment. Paper [7] constructs probabilistic models for mobile robot motion planning as the uncertainties is from the impact of sensor and actuator noise. And in paper [8], by considering the environment in which the dynamic nature comes from doors that can open or close, an MDP is modeled to find a control strategy. However, in all above papers the method of building roadmap is based on the way of partitioning the environment into polygonal regions. It will require the whole divided polygon to have the same properties, which is unfavorable to fully describe the environmental conditions. As a result, in this paper a randomized sampling-based method is used to express the robot free space as a large number of sampling points. Differing from most path planning methods, we apply temporal logic language to describe the task. In the general path planning problem, a mission objective remains simple as given as “go from A to B and avoid obstacles”, where A and B are two regions of interest. However, a mission might require the attainment of either C or D, visiting regions sequentially, or the satisfaction of more complicated temporal and logic conditions about the reachability of regions of interest. So we use Computation Tree Logic (CTL) to meet the demand of expressing tasks diversely.

2. Problem definition and approach

2.1 Problem Definition

The nurse assistant robot working in a hospital environment is the experimental subject in this paper. Nurse assistant robot is a kind of mobile service robots, autonomous navigation in the hospital environment, which is used to deliver drugs and foods to patients to improve the automation level of the ward [9]. The mission scenario is shown in Fig. 1. We simplify some of the rooms and facilities replaced with black areas as obstacles. Some regions in this environment are labeled with properties such as Home, Service, Room and Goal, to distinguish the different areas of interest. In addition, the map is divided into different color regions according to the environmental conditions which are defined by the different parameter values (details in the Section 3). The deeper color indicates the worse environmental condition of current region and also means the stronger influence on the robot motion.

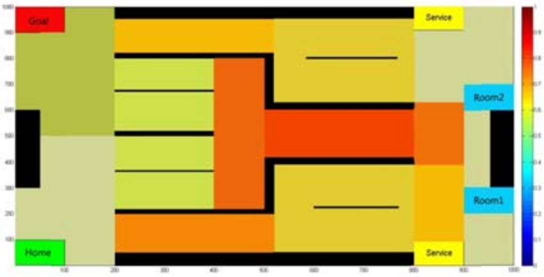


Fig. 1. Mission scenario

We assume that the nurse assistant robot is programmed with a small set of feedback control primitives allowing it to move inside each region and from one region to an adjacent region, but these control primitives are not completely reliable. That means, if at a given region the control primitive is used to take the robot to a specific adjacent region, it is possible the robot will move to a different adjacent region or failure behavior occurred. Of course, the success or failure rates depend on the influencing parameters of environment factors. Based on the above mentioned assumptions, this paper considers the impact of the surrounding environmental factors. Four parameters are taken into account including the size of the active regions, the number of rooms, the number of obstacles in current region and the regional saturation.

2.2 Research Method

The main steps of our method are shown in Fig. 2. First, we use randomized sampling-based method to build the roadmap on the basis of the environment, and translate the influencing parameters into transition probabilities. Then, we construct Markov decision process (MDP) model by probabilistic model checking method and describe complex properties by Probabilistic Computation Tree Logic (PCTL) language [10-11]. In the end, the properties are verified through PRISM tool to guide the robot behavior. The verification results consist of the following two parts. The quantitative data will be obtained including the maximum probability of completing mission and the minimum time to fulfil a task, etc. In addition, the corresponding state sequences satisfying the specification of properties also can be obtained, which can be translated into the path trajectories to generate the optimal control strategy.

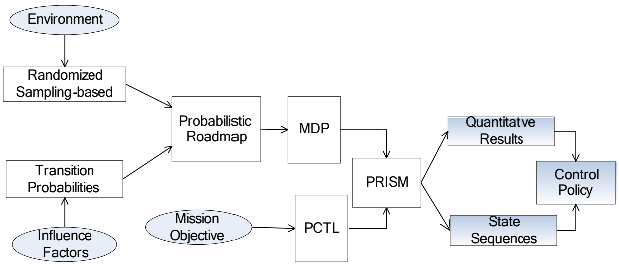


Fig. 2. Research flowchart

However, the sequence of states achieved by probabilistic model checking method is discrete. For path smoothing, a hierarchical planning method is proposed combined with the Probabilistic Road Map (PRM) algorithm in the lower level. The state sequences satisfied the properties are obtained in high level with probabilistic model checking method, which are corresponding to the abstract behaviors of robot moving from one point to its adjacent point. By using PRM algorithm, the low level planner will generate the physical trajectories and drive the motion of the robot through regions.

3. Construction Probabilistic model

3.1 Randomized Sampling-based Method

The roadmap of environment should be built before modeling. A randomized sampling-based method is used to build the environment map with reference to the path planning algorithm PRM [12]. The environment is a two-dimensional space containing a number of obstacles, denoted as Z. The whole environment is removed obstacles and edge regions as free configuration space denoted as Z_{free}, in which robot can free to move. We do not consider the dynamic constraints of mobile robot, regarding it as a point. The main work at this stage is to build a probabilistic roadmap G, which is a random network representing the free space Z_{free}.

Probabilistic roadmap is an undirected graph, denoted as G = (Ga, Ge). Ga is a set of points obtained by random sampling robot positions in Z_{free}. Ge is a set of edges from the local planner which is used to establish connections for each node in Ga by finding their neighbor nodes. Here is the method of local planner.

Firstly, a point is selected from the entire environment. Then we apply the collision detection program to detect the position of this point. If it is in the threatened area, the point will be deleted. If this point belongs to the free space, it will be added to the roadmap.

Try to add the new points to the roadmap, repeating the cycle until the roadmap is complete.

3.2 Probabilistic Analysis

In this section, we will introduce four environmental factors that may affect the behavior of robots. Firstly we consider the size of active areas as a factor that will affect the robot behavior. Nurse assistant robots mostly realize the self-localization by recognizing the doorplate numbers based on the visual navigation combining with RFID technology [13]. In addition, studies show that the greater the number of obstacles in activity area the more efforts robots need to spend to avoid collision. While the regional saturation (crowd level) is larger, it will lead to more serious conflict between the robots and obstacles, or even deadlock occurred. Four environmental factors are shown in Table 1. These parameters contain two discrete variables: the number of rooms n and the size of active area r, $n,r \in [1,2,3]$, where the values are proportional to the room numbers and area sizes. And two continuous variables: the regional saturation $q \in (0.20, 0.80)$ and the number of obstacles $o \in (0,50)$.

Table.1. Environmental Parameters

Factors	Parameters	values		
Room Numbers	n	1	2	3
Active Area Size	r	1	2	3
Obstacle Numbers	o	0-50 (continuousvariable)		
Regional Saturation	q	0.20-0.80 (continuous variable)		

Here are the main steps how to calculate the transition probability from the four parameters in table 1.

1) After the randomized sampling-based method, the environment is divided into a large number of points $Ga = \bigcap_{i=1}^{N} Gai$. Then the values of four parameters corresponding each point are obtained. The whole map is transformed into a set of points which are defined different attributes.

2) Then we sample the robot behaviors under different environmental parameters. As a result, the sample set is formed.

We calculate the transition probability P by the regression equation as shown in Formula.1.

$$\mathrm{P(y=j/x)}=\frac{\exp(a_j+\sum_{k=1}^{4}\beta_{jk}x_k)}{1+\sum_{j=1}^{2}\exp(a_j+\sum_{k=1}^{4}\beta_{jk}x_k)},\ j=1,2,3 \tag{1}$$

a_j and β_{jk} are the regression intercept and regression coefficient calculated by Logistic regression model. x is the explanatory variable as shown in Table 1 and k is the number of explanatory variables. P(y=j/x) is expressed as conditional probability and y is a function of x. j is the category of response variables and j=1,2,3 are the variable values when the robot performs the motion behavior correct, wrong and failure respectively. j=1 is the dependent variable.

3.3 Construction of an MDP Model

We now describe the construction of the MDP. The MDP has the form $\mathrm{M=(S,s_0,Act,R,Steps,L)}$, where:

1) $\mathrm{S=\{L_i,W_i,E_i\}}$ is a finite set of states. Li=$\mathrm{G_{ai}}$ is a subset which is one-to-one correspondence to the nodes in roadmap. $\mathrm{W_i}$ represents an intermediate state that the model reaches by mistake instead of the initial planning state. $\mathrm{E_i}$ is a state that the model reaches when the failure navigation or deadlock is occurred.

2) $Act=\{Goi\_j, Losei\_j\}$ is a set of actions, where i is the subscript of current state and j is the subscript of next state. $Act=Goi\_j$ is the corresponding action of MDP when robot moves from node Gai to Gaj. $Act=Losei\_j$ is the action that MDP model applies when robot moves to other nodes instead of the target.

3) $\mathrm{R}=\mathrm{R}(Ga_i(\mathrm{x}_i,y_i),Ga_j(x_j,y_j))$ is the action cost. In this paper, we define the cost of action is only related to the length between two transferred points.

4) $Steps=T(s,a,s'),a\in Act:T=\{P_{corr},P_{wrong},\ P_{fail}\}$ is the set of probability values corresponding to the motion performed correctly, wrong and failure, respectively. When Act=Goi_j is applied, the value of transition probability is determined by the environmental parameters of current point and the next transferred point as shown in Table 1, denoted as $P_{(i,j)}=P_{corr/wrong/fail}(Ga_i(n_i,r_i,o_i,q_i),Ga_j(n_j,r_j,o_j,q_j))$. In particular, when the wrong motion behavior is happened, Act=Losei_j is applied and the value of $P_{wrong}(i,j,k)$ containing three parameters is determined by the environmental factors of all its neighbors.

5) L= {Home, Goal, Service, Room1, Room2} is the set of labelling.

As an example, consider the environment depicted in Fig. 3. The map is denoted as G = (Ga, Ge), where Ga= {Gai}, i∊[1,6] and Ge contains ten edges as two-way connected. To illustrate how the MDP is modelled, a fragment of MDP is only considered Ga1->Ga2->Ga5 path (from Fig. 3) as shown in Fig. 4. As indicated in the figure, the probability of applying Act=Go1_2 at Ga1 node and ending up at Ga2 when the motion behavior is performed correctly is Pcorr(1,2)=0.867, when the motion behavior is performed wrong is Pwrong(1,2)=0.103, and the probability value is Pfail(1,2)=0.03 when failure is occurred. In addition, when the wrong behavior is happened, Act=Lose1_2 is applied. As a result, adjacent regions Ga3 and Ga4 will be reached with probabilities Pwrong(1,2,3)=0.381 and Pwrong(1,2,4)=0.619. Of particular note is Pwrong(i,j,k) means the probability when the robot want to move from node Gai to Gaj but is wrong arrived at Gak. The value is determined by the environmental factors of all adjacent regions of Gai except Gaj. Actions Go2_5 and Lose2_5 are applied as the same and not explained here.

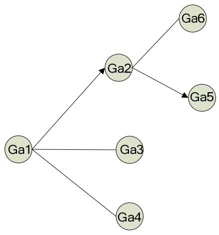


Fig. 3. Branch of roadmap

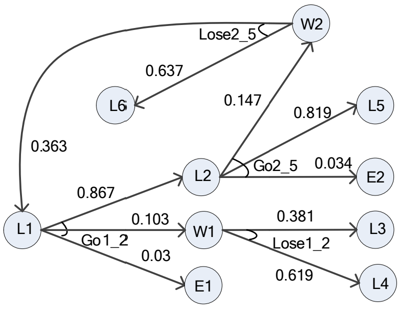


Fig. 4. Fragment of formal model

4. Probabilistic analysis and verification Results

This paper uses PRISM tool developed by the Birmingham University. PRISM tool can support model checking for several types of probabilistic models such as discrete-time Markov chains and Markov decision processes, and adopts a uniform modelling language, for all the probabilistic models supported [14-15].

4.1 Quantitative Results

Prop.1: The maximum probability to complete the mission: "Reach Goal by driving through Room2."

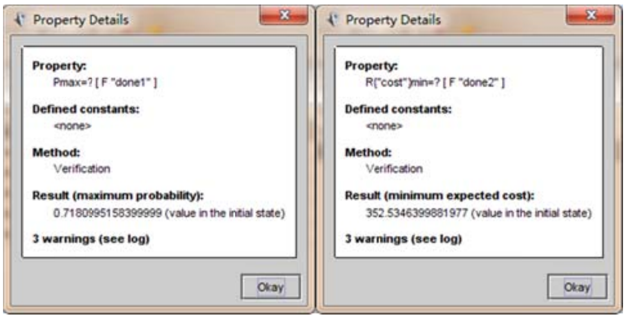
Prop.2: The minimum cost to complete the mission: "Reach Goal by driving through Service."

Properties 1 and 2 translate naturally to PCTL formulas Φ1 and Φ2, respectively, where

$\phi_1: P\max = ?\ [F\ (Room2\ \mathrm{U}\ Goal)]$

$\phi_2: R\{\text{"cost"}\}\min = ?\ [F\ (Service\ \mathrm{U}\ Goal)]$

We assume that the robot is initially at Home. The verification results are shown in Fig. 5. The maximum probability for property Φ1 is 0.718 as shown in Fig. 5(a) and the minimum expected cost for property Φ2 is 352.534 as shown in Fig. 5(b). The simulation showed that the probability of satisfying Φ1 was 0.780 and the minimum expected cost of Φ2 is 361.027. Comparing two groups of data, it can be concluded that the verification results match the simulation data within 5% error range, so these quantitativedata are valid.



Property:
Pmax=? [F "done1"]
Defined constants:
<none>
Method:
Verification
Result (maximum probability):
0.7180995158399999 (value in the initial state)
3 warnings (see log)

Property:
R{"cost"}min=? [F "done2"]
Defined constants:
<none>
Method:
Verification
Result (minimum expected cost):
352.5346399881977 (value in the initial state)
3 warnings (see log)

(a) Result of property 1 (b) Result of property 2

Fig. 5. Verification Results in Matlab

*Prop.3***:** when the regional saturation at region $(x \in [200,500], y \in [0,200])$ is gradually increased, the maximum probability to complete the mission: "Reach Room1 by driving through Service." is shown as Fig. 6. In PCTL the property is expressed by:

$\phi_3: P\max = ?\ [F\ (Service\ \mathrm{U}\ Room1)]$

Before verification, other environmental parameters are set to room numbers n=2, active area r=2, and obstacle numbers o=10. The horizontal axis represents the value of regional saturation. When it is gradually increased, the maximum probability satisfying Φ3 is represented in vertical axis. It indicates that, when other parameters remain unchanged, the higher regional saturation leads to the lower maximum probability as in Fig. 6.

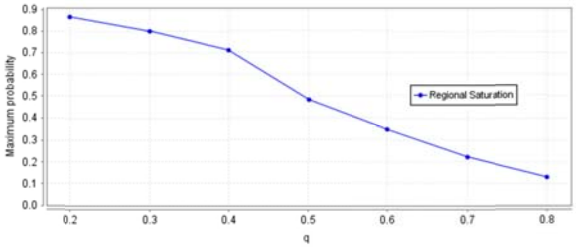


Fig. 6. Result of property 3

4.2 Path Planning

As the cost definition of each action in MDP model is only related to the path length, the optimal path with minimal expected cost can be obtained by verifying relevant properties. The red path in Fig. 7 is the simulation result of verifying Φ2. This path is robot trajectory, which considers the uncertainty, satisfying the minimum expected cost and meeting the need of property Φ2. In order to verify the correctness of this method, we use Dijkstra algorithm in VC6.0 to find the shortest path on the map in Fig. 7. The results show that the nodes outputted by this algorithm and the nodes on the red path in Fig. 7 are consistent with 93.1%.

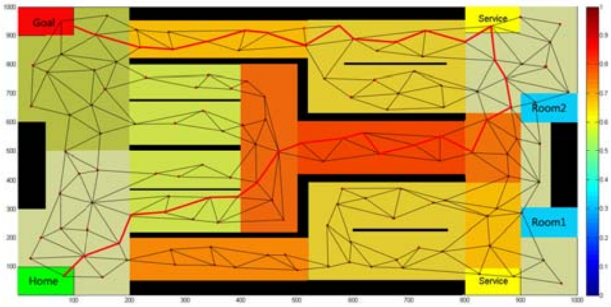


Fig. 7. Path of property 2

*Prop.4:*The maximum probability to complete the task: "Reach Goal by passing one of Service areas among driving through Room1 then Room2." Properties 4 translates naturally to PCTL formulas Φ4, where,

ϕ_4: $P\max = ?\ [F\,(Service \wedge (Room1 \wedge (F\ Room2)))\ \mathrm{U}\, Goal)]$

The verification result is the red path shown in Fig. 8.Now we modify the value of parameters as the regional saturation is increased five percent in region $(x \in [200,500], y \in [0,200])$ and ten percent in the region $(x \in [600,800], y \in [450,600])$ is reduced. Then we verify the property Φ4 again and the experiment result is the blue trajectory in Fig. 8. Comparing the two paths, it can be concluded that the planning path will be different when the external environment is changed, but the unchanged is the generated trajectories will always try to avoid the hostile environments for maximum probability to complete the mission. In addition, the property requires the planning path must go through some Service region. As a result, the blue trajectory chooses the Service in the above instead of the bottom one in order to shorten the path length to ensure the robot completes task with the maximum probability.

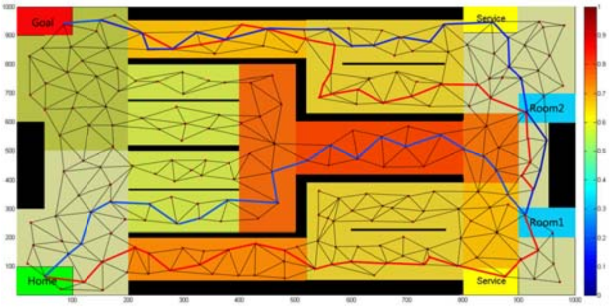


Fig. 8. Path of property 4

5. Dynamic obstacle avoidance strategies

In the previous section, the plan by the probability of detection methods meet global path attributes to ensure minimal impact on the robot. But in the process of moving the robot might encounter at any direction and trajectory of dynamic obstacles are not clear, so the robot motion process each line further, we must continue to use environmental information within the sensor detection field of view, and to distinguish the static obstacles and dynamic obstacles, if detected dynamic obstacles, you need to detect its direction, speed and trajectory, and then use the appropriate method for dynamic obstacle avoidance [7].

In the robot's motion environment, due to the static obstacles easily coordinate mapping, is generally regarded as known information, but it is uncertain dynamic obstacle information. The purpose of this section is to ensure that the path planning robot collision avoidance under the premise to meet the optimum path along a moving target task. The basic principle of dynamic obstacle avoidance in this paper is to perform global path planning by a known environmental information, and real-time obstacle avoidance way of combining operations. Rely on internal geometry or topology robot saved maps and other information about the environment, on the basis of pre-planning a route on a global, dynamic obstacle avoidance method for robot navigation.

Assumption that the robot at any time based on the detected current position as environmental information center, r is the radius of the region, the rate of movement of the robot is Vr; robot in two-dimensional plane distribution of the static obstacles finite and limited dynamic obstacle objects, dynamic trajectory and direction of the obstacle is unknown, but assuming that all movement locus of the moving obstacle is unidirectional, non-self-intersecting smooth trajectory, and within the detection range of the robot is approximated as a linear motion direction of the same .

The specific method is divided into three steps: Scene forecasts, collision detection and collision avoidance local detour, described in detail below.

1) Scene forecast period: continuous detection robot obstacle information within the field of view, if detected in time ti dynamic obstacles, measured obtaining the coordinate position, and then after a short time of ∆t measured position. Then, using Equation (2) estimating the moving speed of the obstacle, using the equation (3) the calculation of the field of view obstacles the robot trajectory is approximately linear.

$$v = d(p_d(x(t+\Delta t), y(t+\Delta t), p_d(x(t), y(t)))) / \Delta t \quad (2)$$

The linear equation y = kx + c, where

$$\begin{cases} k = (y(t+\Delta t) - y(t) / (x(t+\Delta t) - x(t)) \\ c = y(t) - x(t) / (x(t+\Delta t) - x(t)) \end{cases} \quad (3)$$

2) Collision detection phases: collision safety distance threshold hypothesis D is considered an obstacle to the robot and dynamic volume obtained after informed by Step one obstacle at the moment ti trajectory moves along at a speed of formula (3), you can predict it in the location and coordinates different times,, k = 1,2,3 ... N. Then, we follow the movement and speed Vr step Lr virtual robot forward, time tr reach the edge of the visible range of the robot, so, write T = {ti, ti + ∆ t, ti + 2 ∆ t, ..., tr}, robot location of the collection is a virtual walk. Finally, according to the formula (4) is determined in the process of moving the robot at any time ∆t is a collision occurs, the condition satisfies the equation, denoted by P(tc), led by the collision prediction point. Arbitrary, detection formula as (4), if a collision is forecast to be carried out in accordance with Step Three collision avoidance detour local planning.

$$\begin{aligned} & d(P(t), P_d(t)) > D, t < t_r, t \in T \\ & d(P(t), P_d(t)) \le D, t = t_r, t \in T \end{aligned} \quad (4)$$

3) Local Bypass collision avoidance stage: This stage is mainly to avoid dynamic obstacles and take the local planning stages of a temporary planning a detour path, when the robot avoid obstacles will go back to the original dynamic planned global on the path. Collision avoidance planning method steps, the prediction obtained in step two collision points recorded as a collection of Co, and in order to prevent the error affected the collision point of the collection to the former path, after moving a point respectively; then modify the collection the value of all property as an obstacle collision point,

is considered prohibited by region; finally re-select the method in accordance with 3.1 path in local static environment.

Of particular note, the paper constructs a probability map for each node element is assigned a value corresponding parameters according to the external environmental conditions, the properties of the node according to the parameters specified value and where the regional context, so when bypass local planning occurs, it will be discarded node preference and point the same attributes as the detour choice. Suppose node Li is recorded as predicted collision point, the model will search and Li attributes the same around other non-collision node, and then modify its variable values, increasing migration probability of the node arrives, thereby ensuring that as much as possible under the premise of collision avoidance does not change the overall direction of the path, the maximum probability accomplish a task. The detour robot collision avoidance strategies will ultimately return to the global path planning, to ensure the safety of robot motion paths and meet the optimum, improve work efficiency of the robot.

In order to verify the effectiveness of this method, we conducted a related experiment, the experimental results shown in Fig. 7. The figure at the top of the black solid line shows the trajectory of dynamic obstacles, red thread below the global path planning in order to verify the properties Φ4 get red thick line indicates the local detour route planning and obstacle avoidance.

Fig. 9, black node L3, L4, L5 is predicted collision point robot with dynamic obstacles, we can see, in order to avoid obstacles, by passing the local path planning collision prediction point L3, L9 selected node as a detour . Probabilistic model defined herein in L2, L3, L8, L9 four nodes having the same properties, node L3 is set to predict the collision point, modify its variable value, then L8 and L9 node as the preferred detour nodes, but in compared to the node model L8 L9 having a more favorable environment parameter (defined in Table 1), the probability of migration path is high, the local scheduler node selected as the detour node L9. Effective path seen from the results, even if the situation encountered dynamic obstacles, we propose a method to avoid collision by local planning can avoid obstacles, select optimum path and eventually come back to the initial global path planning to meet the property requirements and to ensure that the path is better.

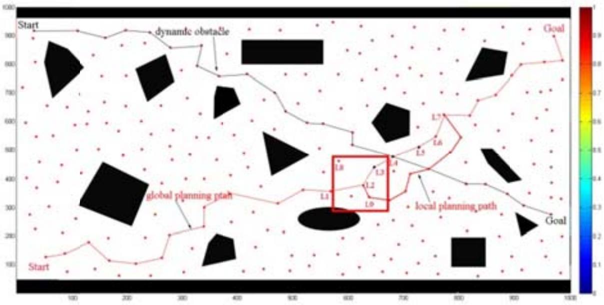


Fig. 9. Dynamic obstacle avoidance experiment

6. Summary

This paper provided an approach to solve the path planning problem for mobile robot affected by their surroundings. We presented a method obtaining the transition probabilities from the environmental parameters, by which the robot behavior became an uncertain action according with practical applications. The motion of the mobile robot has been modeled as an MDP and adopted probabilistic model checking method to find the optimal path trajectory that maximizes the probability of satisfying the specification as a PCTL formula. And the quantitative verification results have been generated which provided a useful reference for the path choice.

In future work, we plan to apply the probabilistic model checking to the mobile robots moving in more complex environment, and refine probability factors by considering the diversity of robots behavior to complete path planning problem.

References

[1] C. Wang, Y. C. Soh, and H. Wang, A hierarchical genetic algorithm for path planning in a static environment with obstacles, IEEE Conference on Electrical and Computer Engineering, Canada, vol. 3, p. 1652-1657, 2002.

[2] J. J. Kuffner and S. M. LaValle, RRT-connect: An efficient approach to single-query path planning. Robotics and Automation, Proceedings of the IEEE International Conference on Robotics and Automation, 2000, vol. 2, p. 995-1001.

[3] G. E. Fainekos, H. K. Gazit and G. J. Pappas, Hybrid controllers for path planning: A temporal logic approach, IEEE Conference on Decision and Control, 2005, p. 4885-4890.

[4] H. K. Gazit, G. Fainekos, and G. J. Pappas, Where's Waldo? Sensor-based temporal logic motion planning, Proceedings of IEEE International Conference on Robotics and Automation, 2007, p. 3116–3121.

[5] X. C. Ding, A. Pinto, and A. Surana, Strategic planning under uncertainties via constrained Markov Decision Processes, IEEE International Conference on Robotics and Automation, 2013, p. 4568-4575.

[6] I. Cizelj, X. C. Ding, and M. Lahijanian, Probabilistically safe vehicle control in a hostile environment, arXiv preprint arXiv:1103.4065, 2011.

[7] M. Lahijanian, J. Wasniewski, and S. B. Andersson, Motion planning and control from temporal logic specification with probabilistic satisfaction guarantees, Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, p. 3227-3232.

[8] M. Ayala, A. I., S. B. Andersson, and C. Belta, Temporal logic control in dynamic environments with probabilistic satisfaction guarantees, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, p: 3108-3113.

[9] L. Q. Wang, Q. X. Meng and L. B. Guo, Study on nurse assistant robot, China Medical Devices Information, 2003, vol. 9, no. 4, p. 21-23.

[10] Ciesinski F, Größer M. On probabilistic computation tree logic, Validation of Stochastic Systems. Springer Berlin Heidelberg, 2004 p : 147-188.

[11] D. Parker, Implementation of Symbolic Model Checking for Probabilistic Systems, University of Birmingham, 2002.

[12] Bohlin, Robert, and E. E. Kavraki, Path planning using lazy PRM, Proceedings of the IEEE International Conference on Robotics and Automation, 2000, vol. 1, p. 521-528.

[13] J. Zhang, G. S. Zhang and Z. G. Bing, The navigation system design of mobile robot for hospitals, Microcomputer Information, 2007, vol. 23, no. 20, p. 200-202.

[14] A. H, M. Kwiatkowska, G. Norman and D. Parker, PRISM: A Tool for Automatic Verification of Probabilistic Systems, In Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06), 2006, p. 441-444.

[15] M. Kwiatkowska, G. Norman, and D. Parker, Probabilistic symbolic model checking with PRISM: A hybrid approach, International Journal on Software Tools for Technology Transfer, 2004, vol.6, no.2, p.128-142.