Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

On-line real-time path planning of mobile robots in dynamic uncertain environment

ZHUANG Hui-zhong, DU Shu-xin‡, WU Tie-jun

(*Institute of Intelligent Systems and Decision Making, National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China*)

E-mail: hzzhuang@iipc.zju.edu.cn; shxdu@iipc.zju.edu.cn; tjwu@iipc.zju.edu.cn

Received Mar. 10, 2005; revision accepted July 15, 2005

Abstract: A new path planning method for mobile robots in globally unknown environment with moving obstacles is presented. With an autoregressive (AR) model to predict the future positions of moving obstacles, and the predicted position taken as the next position of moving obstacles, a motion path in dynamic uncertain environment is planned by means of an on-line real-time path planning technique based on polar coordinates in which the desirable direction angle is taken into consideration as an optimization index. The effectiveness, feasibility, high stability, perfect performance of obstacle avoidance, real-time and optimization capability are demonstrated by simulation examples.

Key words: Mobile robot, Dynamic obstacle, Autoregressive (AR) prediction, On-line real-time path planning, Desirable direction angle

doi:10.1631/jzus.2006.A0516 **Document code:** A **CLC number:** TP24

INTRODUCTION

Path planning of mobile robots is one of the key issues in robotics research on the problem of a robot finding a collision-free path from beginning to goal in the presence of obstacles. Depending on the environment surrounding the robot, it can be classified as follows:

(1) Path planning for static obstacles in completely known environment;

(2) Path planning for static obstacles in unknown or partially known environment;

(3) Path planning for dynamic obstacles in completely known environment;

(4) Path planning for dynamic obstacles in unknown or partially known environment.

To the former two cases, that is, path planning in the presence of static obstacles, there are various approaches, such as ℂ-space method and artificial potential field method. They may obtain perfect results under some certain condition. But in fact, robots mostly work in dynamic uncertain environment that includes static obstacles with unknown position and dynamic obstacles with uncertain trajectory. So the collision-free navigation of mobile robots in dynamic uncertain environment is so complex that it is still an intractable topic by far.

Many real-time navigation systems have been developed recently. Fujumura (1992) proposed an algorithm for obstacles with known trajectories. Tang *et al.*(2000) treated instantaneous dynamic obstacle as static and proposed an optimal path based on a grid method called dynamic grid method. Fiorinio and Shiller (1998) advanced a view of velocity obstacle that plans on-line motion based on relative velocity between the robot and moving obstacles. Rude (1997) used Time-Space coordinates to translate robot's path planning in 2D dynamic environment into static path planning in 3D space. Takeshi (1994) used three-tier fuzzy control method to

‡Corresponding author

adjust robot's motion direction and velocity for detouring dynamic obstacles.

It is difficult to give obstacle's reliable motion information obviously, e.g. velocity and direction. In fact, such real-time information is hidden in next-time obstacle's position. So we can obtain obstacle's motion information provided that next-time the obstacle's position is predicted by some prediction algorithm. Some navigation systems combine path planning with obstacle motion prediction. A prediction model takes object's previous information and uses various methods to predict the motion trend of the object to get its next position. In motion prediction, numerical prediction approaches, such as curve fitting or regression methods (Sen and Srivastava, 1990), are widely used because they are simple and convenient. Other models that may be used to improve the prediction results are hidden Markov stochastic models (Zhu, 1991), the Grey prediction (Luo and Chen, 1999), etc. The Kalman Filter (Kalman, 1960) is also used for predicting future positions and orientations of a moving object in dynamic environments. The Kalman Filter's recursive nature enables it estimate a real-time system state.

The motion trajectories of moving obstacles are predicted by using an autoregressive (AR) model in this paper. Compared with other models above, the AR model is so simple and its algorithm is so fast that the robot can give a quick response when encountering obstacles. In the meanwhile, prediction precision can also be guaranteed. Positions of moving obstacles are sampled by the robot's sensor system. The sampling position information on dynamic obstacles is treated as instantaneously static. With current sampling positions, the AR model predicts future obstacle's positions in the next sampling duration. Then the robot's motion path is planned with such predicted positions. Thus, dynamic collision-free path planning is translated into static one. Combining global planning with local planning, we propose an on-line real-time path planning technique based on polar coordinates in which the desirable direction angle is taken into consideration as an optimization index. Detecting unknown obstacles with local feedback information by the robot's sensor system, this approach orients the desirable direction of the mobile robot so as to generate local sub-goal in every planning window. As a result, the difference between real direction angle and desirable direction angle of robot motion steers the mobile robot to detour collisions and advance toward the target without stopping to re-plan a path when new sensor data become available.

Trajectory prediction of moving obstacles is elaborated on in Section 2. On-line real-time path planning of mobile robots in dynamic uncertain environment is proposed in Section 3. The effectiveness, feasibility, high stability, perfect performance of obstacle avoidance, real-time and optimization capability of the proposed approach are demonstrated by simulation examples in Section 4. And Section 5 provides the results of our study.

TRAJECTORY PREDICTION OF DYNAMIC OBSTACLES

In our analysis, we consider a 2D navigational space. However, the framework discussed here can be readily extended to 3D or higher dimensional space. As depicted in Fig.1, suppose a robot R is to travel from starting point S to goal point G in a navigational space consisting of M moving obstacles $O_1,O_2,\ldots,O_M$. Clearly, if there are no moving obstacles, the straight-line path $\overline{SG}$ is a reasonable solution provided that no obstacle is located along $\overline{SG}$. When the obstacles are in motion, the straight-line path $\overline{SG}$ is not necessarily a collision-free path and a better solution must be looked for.

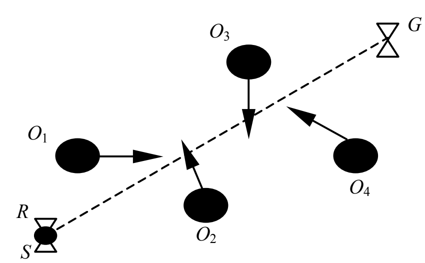

Fig.1 Robot's workshop in the presence of moving obstacles

AR modeling of the obstacle's position

In the following sections, we develop a prediction model to be used by a robot to decide about future positions of moving obstacles. Our intention is to use this model within a trajectory planning algorithm in a time-varying environment. Before the robot starts to interact with its environment, data on visible obstacles are collected, via sensors, for a short period of time. This step enables the robot to learn about any moving obstacles in its visibility field at discrete points in time-space. Therefore, statistical modelling using difference equations is appropriate for predicting future position of a moving obstacle based upon its previous positions. But since sensory readings are usually noisy, an autoregressive (AR) model is more relevant and useful.

Let the sensing system of the robot sample an obstacle O_k's actual position, $\boldsymbol{p}_k(t)$, at time t=1,2,…, where $\boldsymbol{p}_k(t)=(\boldsymbol{X}_k(t),\boldsymbol{Y}_k(t))^{\mathrm{T}}$. And $\hat{\boldsymbol{p}}_k(t)$ defines the predicted position of O_k at time t. The sequence of positions $\{\boldsymbol{p}_k(t), t=1,2,\ldots\}$ is fitted to a nth order AR model described by the following difference equation:

$$\boldsymbol{p}_k(t)=\sum_{i=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{p}_k(t-i)+\boldsymbol{e}(t), \tag{1}$$

where $\boldsymbol{\alpha}_i$ (i=1,…,n) are AR coefficients, $\boldsymbol{e}(t)$ denotes the prediction error. Depending on different applications or assumptions, the coefficients $\{\boldsymbol{\alpha}_i, 1<i\le n\}$ may be scalars or matrices. For instance, in the case of a mobile robot, the coefficients are (2×2) matrices.

If the obstacle O_k undergoes slowly changing acceleration (small positional sampling duration), it is reasonable to model accelerations of O_k, $\{\boldsymbol{a}_k(t), t=1,2,\ldots\}$, with a first order AR model as follows:

$$\boldsymbol{a}_k(t)=\boldsymbol{\beta}_{k,t}\boldsymbol{a}_k(t-1)+\boldsymbol{w}(t), \tag{2}$$

with $\boldsymbol{w}(t)$ as the prediction error. Note that for different assumptions, the coefficient $\boldsymbol{\beta}_{k,t}$ can be a scalar, a diagonal or general matrix. For instance, if the motions along x and y are correlated, $\boldsymbol{\beta}_{k,t}$ will be a general matrix. In each case, $\boldsymbol{\beta}_{k,t}$ can be either dependent or independent of time t. If $\boldsymbol{\beta}_{k,t}$ is independent of time, it is only required to compute it once. In general, $\boldsymbol{\beta}_{k,t}$ is estimated adaptively as new measurements are made available by the sensing system. In terms of the obstacle's velocity and position, the robot's acceleration can be calculated by:

$$\begin{aligned}\boldsymbol{a}_k(t)&=\boldsymbol{v}_k(t)-\boldsymbol{v}_k(t-1)\\&=|\boldsymbol{p}_k(t)-\boldsymbol{p}_k(t-1)|-|\boldsymbol{p}_k(t-1)-\boldsymbol{p}_k(t-2)|\\&=\boldsymbol{p}_k(t)-2\boldsymbol{p}_k(t-1)+\boldsymbol{p}_k(t-2),\end{aligned} \tag{3}$$

where $\boldsymbol{v}_k(t)$ represents velocity at time t. Substituting Eq.(3) in Eq.(2), we have:

$$\begin{aligned}&\boldsymbol{p}_k(t)-(2+\boldsymbol{\beta}_{k,t})\boldsymbol{p}_k(t-1)+(2\boldsymbol{\beta}_{k,t}+1)\boldsymbol{p}_k(t-2)\\&\quad-\boldsymbol{\beta}_{k,t}\boldsymbol{p}_k(t-3)-\boldsymbol{w}(t)=0,\end{aligned}$$

that is,

$$\begin{aligned}\boldsymbol{p}_k(t)&=(2+\boldsymbol{\beta}_{k,t})\boldsymbol{p}_k(t-1)-(2\boldsymbol{\beta}_{k,t}+1)\boldsymbol{p}_k(t-2)\\&\quad+\boldsymbol{\beta}_{k,t}\boldsymbol{p}_k(t-3)+\boldsymbol{w}(t).\end{aligned} \tag{4}$$

Estimating the model coefficients

Let us first assume that $\boldsymbol{\beta}_{k,t}$ is a scalar and independent of time t, $\boldsymbol{\beta}_{k,t}=\boldsymbol{\beta}_k$; also there is a batch of position measurements available, $\{\boldsymbol{p}_k(t), 1\le t\le N\}$. To estimate $\boldsymbol{\beta}_k$, first the acceleration sequence $\{\boldsymbol{a}_k(t), 3\le t\le N\}$ is formed from Eq.(3). Then, a first order AR model is fitted to this sequence in a least squares sense, that is, $\hat{\boldsymbol{\beta}}_k$ is chosen such that

$$\hat{\boldsymbol{\beta}}_k=\arg\min_{\boldsymbol{\beta}_k}\sum_{i=4}^{N}[\boldsymbol{a}_k(i)-\boldsymbol{\beta}_k\boldsymbol{a}_k(i-1)]'[\boldsymbol{a}_k(i)-\boldsymbol{\beta}_k\boldsymbol{a}_k(i-1)]. \tag{5}$$

The solution to the above least squares problem is given by Makhoul (1975):

$$\hat{\boldsymbol{\beta}}_k=\frac{\sum_{i=4}^{N}\boldsymbol{a}_k'(i)\boldsymbol{a}_k(i-1)}{\sum_{i=4}^{N}\boldsymbol{a}_k'(i-1)\boldsymbol{a}_k(i-1)}. \tag{6}$$

The above result takes the following general form when $\boldsymbol{\beta}_k$ is a matrix,

$$\hat{\boldsymbol{\beta}}_k=\left[\sum_{i=4}^{N}\boldsymbol{a}_k(i)\boldsymbol{a}_k'(i-1)\right]\left[\sum_{i=4}^{N}\boldsymbol{a}_k(i-1)\boldsymbol{a}_k'(i-1)\right]'. \tag{7}$$

At this point, we consider the case when $\boldsymbol{\beta}_{k,t}$ changes with time t. In this case, an adaptive algorithm is desired where $\boldsymbol{\beta}_{k,t}$ is updated as a new measurement is acquired by the sensing system. The algorithm proposed by Lee *et al*.(1981) and Shensa (1981) is adopted to determine $\boldsymbol{\beta}_{k,t}$ as follows:

$$\hat{\boldsymbol{\beta}}_{k,t} = \arg\min_{\boldsymbol{\beta}_{k,t}} \sum_{i=4}^{t} \left\{ \lambda^{t-i} [\boldsymbol{a}_k(i) - \boldsymbol{\beta}_{k,t}\boldsymbol{a}_k(i-1)]' \times [\boldsymbol{a}_k(i) - \boldsymbol{\beta}_{k,t}\boldsymbol{a}_k(i-1)] \right\}, \tag{8}$$

where λ ($0<\lambda\leq1$) is an exponential weighting factor. For slowly changing acceleration, λ is close to 1. The solution for this problem is:

$$\hat{\boldsymbol{\beta}}_{k,t} = \frac{\boldsymbol{\Delta}_{k,t}}{\boldsymbol{\eta}_{k,t}}, \tag{9.1}$$

where

$$\boldsymbol{\Delta}_{k,t} = \lambda\boldsymbol{\Delta}_{k,t-1} + \boldsymbol{a}_k'(t)\boldsymbol{a}_k(t-1), \tag{9.2}$$

$$\boldsymbol{\eta}_{k,t} = \lambda\boldsymbol{\eta}_{k,t-1} + \boldsymbol{a}_k'(t-1)\boldsymbol{a}_k(t-1). \tag{9.3}$$

The above result takes the following general form when $\boldsymbol{\beta}_{k,t}$ is a matrix:

$$\hat{\boldsymbol{\beta}}_{k,t} = \frac{\boldsymbol{\Delta}_{k,t}}{\boldsymbol{\eta}_{k,t}}, \tag{10.1}$$

where

$$\boldsymbol{\Delta}_{k,t} = \lambda\boldsymbol{\Delta}_{k,t-1} + \boldsymbol{a}_k(t)\boldsymbol{a}_k'(t-1), \tag{10.2}$$

$$\boldsymbol{\eta}_{k,t} = \lambda\boldsymbol{\eta}_{k,t-1} + \boldsymbol{a}_k(t-1)\boldsymbol{a}_k'(t-1). \tag{10.3}$$

Once the estimation is found for $\boldsymbol{\beta}_{k,t}$, the future positions of O_k are predicted as follows:

$$\hat{\boldsymbol{p}}_k(t+1) = \boldsymbol{p}_k(t) + \boldsymbol{v}_k(t) + \hat{\boldsymbol{\beta}}_{k,t}\boldsymbol{a}_k(t). \tag{11}$$

REAL-TIME PATH PLANNING OF MOBILE ROBOTS

As to on-line real-time path planning of mobile robots in dynamic uncertain environment, we treat every position in the motion trajectory of moving obstacle as instantaneously static. Thus in each sampling duration, current position information of moving obstacles is sampled by the robot's sensor system. Then the AR model elaborated on in Section 2 is used to predict next-time position of moving obstacles. And moving obstacles on these predicted positions are treated as static in that time. As a result, we can implement collision-free path planning by means of an on-line real-time path planning technique based on polar coordinates in which the desirable direction angle is taken into consideration as an optimization index.

Path planning based on polar coordinates space

Consider the 2D path planning for mobile robot in which the robot is shrunk to a single moving point while the immovable obstacles are expanded to circles. Two sets of coordinates are used in our path planning approach discussed in this paper, where global environment coordinates are Cartesian coordinates and local robot coordinates are polar coordinates. The environment Cartesian coordinates absolutely show the positions of starting point, goal point and the real-time location of the mobile robot so that the global information can be considered in every-time planning window. To be convenient, we define the starting point as the original point of Cartesian coordinates. The robot polar coordinates move along with the robot. Current real-time location of the robot is the pole of the polar coordinates and the direction of which robot current location points at the goal is the direction of the polar axis.

Introduction of robot polar coordinates is convenient for describing and computing the motion direction angles of the robot as follows. Since the robot coordinates change at any moment in the motion process, if Cartesian coordinates are adopted, coordinate transform will greatly increase planning time. Moreover, Cartesian coordinates cannot visually express the motion direction angle of the robot. Instead, polar coordinates can be greatly convenient for computing the desirable direction angle. In addition, introduction of environment coordinates is convenient for computing current polar coordinates of the robot and determining current desirable motion direction of the robot according to the goal position in environment coordinates.

To describe conveniently, as shown in Fig.2, we only consider the case of including one obstacle. The starting point S is the original point of the environment Cartesian coordinates and the starting pole of robot polar coordinates. Point G is the goal point. The shadow circle represents the obstacle. Its direction angle is $\varphi(t)$, that is, the real-time angle between the motion direction of the robot and the polar axis direction of current polar coordinates. The motion direction that we want to plan in every local planning is the desirable direction of the robot. And the angle between the desirable direction of the robot and the polar axis direction of current polar coordinates is the desirable direction angle, namely, $\varphi_d(t)$ ($0\leq\varphi_d(t)<2\pi$).

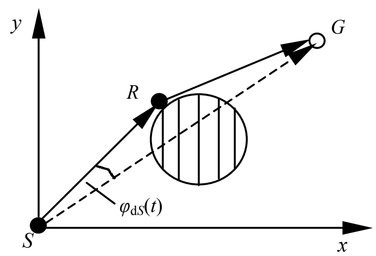

Fig.2 On-line real-time path planning model for mobile robots

According to the model above, the difference between real direction angle and desirable direction angle of robot motion steers the mobile robot to avoid collisions and advance toward the target. Consider a quadratic index

$$E(t)=\frac{1}{2}[\varphi(t)-\varphi_d(t)]^2, \tag{12}$$

where $\varphi_d(t)$ is the desirable direction angle for navigating the robot to the goal, or for avoiding obstacles.

If the robot is required to move toward the desirable direction, then the following control integral law is proposed:

$$\dot{\varphi}(t)=-\eta[\varphi(t)-\varphi_d(t)], \tag{13}$$

where η is a positive constant. The gradient descent method is then used to determine the direction angle to achieve the minimum $E(t)$ implying that the robot is finally navigated to the desirable direction, that is, the direction toward the goal or to avoid obstacles.

To obtain desirable direction angle, a local path planning is executed with one step of the robot. In every-time planning window, the starting point is the end point of the last planning, the current position of the robot, as well as the pole of new polar coordinates. Simultaneously, real-time environment information is detected by the robot's sensor system. According to such environment information and the information on the goal, next motion direction of the robot is planned by every local planning with certain optimization index, that is, determines the desirable direction. Before the next sampling period comes, Eqs.(12) and (13) will drive the robot to move toward the desirable direction. It is repeated as above until the robot reaches the goal. In this strategy of local path planning, the optimization index is the minimum angle (absolute value) between the motion direction of the robot and the polar axis direction of current polar coordinates, that is, the desirable direction angle $\varphi_d(t)$ is minimum.

In local planning, the optimization index is the minimization of the direction angle. Here Fig.2 shows the desirable direction angle in single obstacle case. To figure conveniently, it is assumed that the robot has detected an obstacle in the way towards the goal by sensors at starting point S. And compared with collision-free direction of the other side, the angle between direction SR and current polar axis direction is smaller. So current desirable direction angle is $\varphi_{dS}(t)$. When the robot moves at point R, since there is no obstacle detected in the direction towards the goal, current desirable direction superposes the polar axis direction of current polar coordinates, that is, $\varphi_{dR}(t)=0$. Minimum as the desirable direction angle, it is guaranteed that the planning path is optimal or sub-optimal. This can be proved by the knowledge of plane geometry. Limited by the paper length, the detailed demonstration is skipped here. So index-angle minimum can overcome the drawback in that the local path planning cannot optimize the motion path.

Strategy of obstacle avoidance

The mobile robot discussed in this paper primarily obtains obstacle information by distance sensors. The robot should know obstacle information in forward, left and right directions to determine the next step when moving in collision environment. We assume that the robot moves only in the forward direction. As a result, we can use semicircular range detected by sensors as every-time planning window. In each planning window, let d_f, d_l, d_r be forward, left and right directions respectively, where d_f represents the straight direction towards the goal and d_l, d_r respectively represent any direction in left and right sectors bounded by d_f. Thus, obstacle avoidance of the robot can be divided into the three following cases:

(1) If sensors detect no obstacle along d_f direction, the robot will move towards the goal along d_f direction wherever left or right side has obstacles. At this time, desirable direction angle $\varphi_d(t)=0$, as shown in Figs.3a~3c;

(2) If sensors detect obstacles along d_f direction, then let d_l, d_r be the left and right tangential direction between the robot and obstacle circle. And the angle between d_l, d_r and d_f is β_l, β_r respectively. The robot will move along 'angle minimum' tangent direction, that is, when $\beta_l<\beta_r$, the robot will move along d_l direction and current desirable direction angle $\varphi_d(t)=\beta_l$;

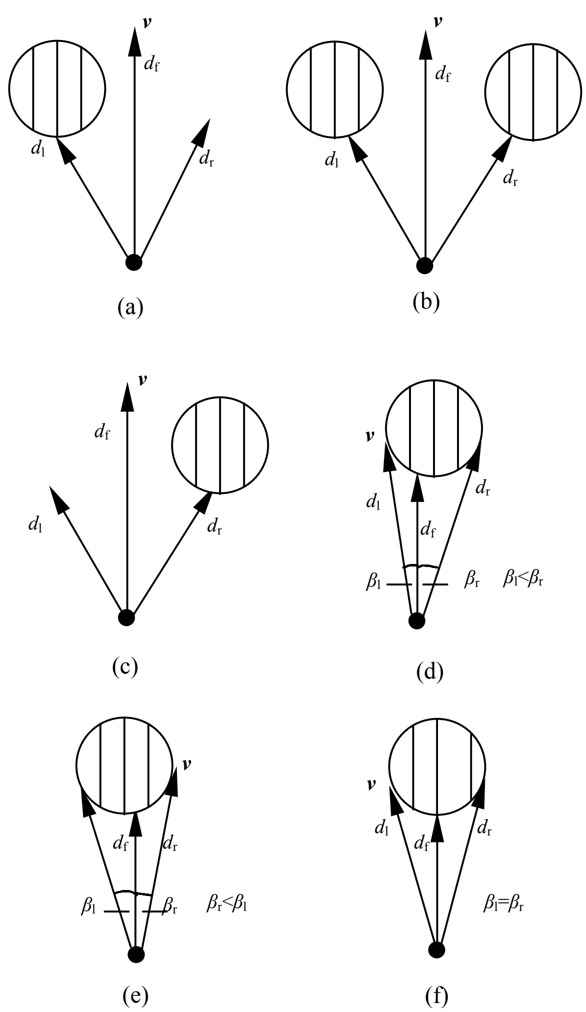

Fig.3 (a)~(c) Case 1; (d)~(e) Case 2; (f) Case 3

when $\beta_r<\beta_l$, the robot will move along d_r direction and current desirable direction angle $\varphi_d(t)=\beta_r$, as shown in Figs.3d~3e;

(3) When $\beta_l=\beta_r$, a special example of Case 2, we prescribe that the robot moves along d_l direction, that is, desirable direction angle $\varphi_d(t)=\beta_l$, as shown in Fig.3f.

Algorithm of real-time path planning

Now an algorithm of on-line real-time path planning for mobile robot in dynamic uncertain environment is presented as follows:

(1) Initialize such parameters as starting point, goal point, work environment, vision field and velocity of the robot;

(2) If the robot reaches the goal, the planning is over;

(3) Update environment information in current planning window according to real-time local detection information;

(4) Are there dynamic obstacles in current planning window? If so, next-time positions of these dynamic obstacles are predicted by the AR model;

(5) According to predicted positions of dynamic obstacles and positions of other static obstacles, desirable direction angle of the robot is determined by the strategy of obstacle avoidance, that is, generating local sub-goal;

(6) The difference between real direction angle and desirable direction angle of robot motion drives the robot to step towards local sub-goal;

(7) Return to Step (2).

Algorithm's reachability and security

In dynamic uncertain environment, if the following theorems are satisfied, the reachability and security of the above algorithm can be guaranteed.

Theorem 1 (Reachability) $\forall P_S$, $P_G\in FD$, the above algorithm must guarantee that the robot moves from the beginning P_S to the goal P_G in limited time, where FD is the feasible domain.

Theorem 2 (Security) When the robot is executing local planning, it is always guaranteed for the robot to move safely if formula $\boldsymbol{v}_R/\boldsymbol{v}_{OH}\geq(L/2+\varepsilon)/(r-\varepsilon)$ is satisfied, where $\boldsymbol{v}_R$, r, ε respectively represent velocity, vision radii and step of the robot and $\boldsymbol{v}_{OH}$, L respectively represent maximal velocity and size of the moving obstacle.

As shown in Theorem 2:

(1) When $\varepsilon \geq r$, collision-free condition cannot be guaranteed;

(2) As to the robot, the faster $\boldsymbol{v}_R$, the larger r and the smaller ε, the more favorable for the robot to avoid obstacles;

(3) The robot's velocity is always slower than that of moving obstacles provided that the robot has the larger vision as indicated by $r>(L/2+2\varepsilon)$.

Such two theorems above can be proved by mathematical induction. Limited by the paper length, they are not proved in detail here.

SIMULATIONS

Several simulation examples were used to demonstrate the models and planning algorithm described above. Fig.4 and Figs.5a~5c show planning results in which the robot detours dynamic obstacle by linear, parabolic, sinusoidal and stochastic motion. In addition, Fig.5d shows robot's collision-free path planning results in dense dynamic uncertain environment that includes static obstacles with unknown position and dynamic obstacles with

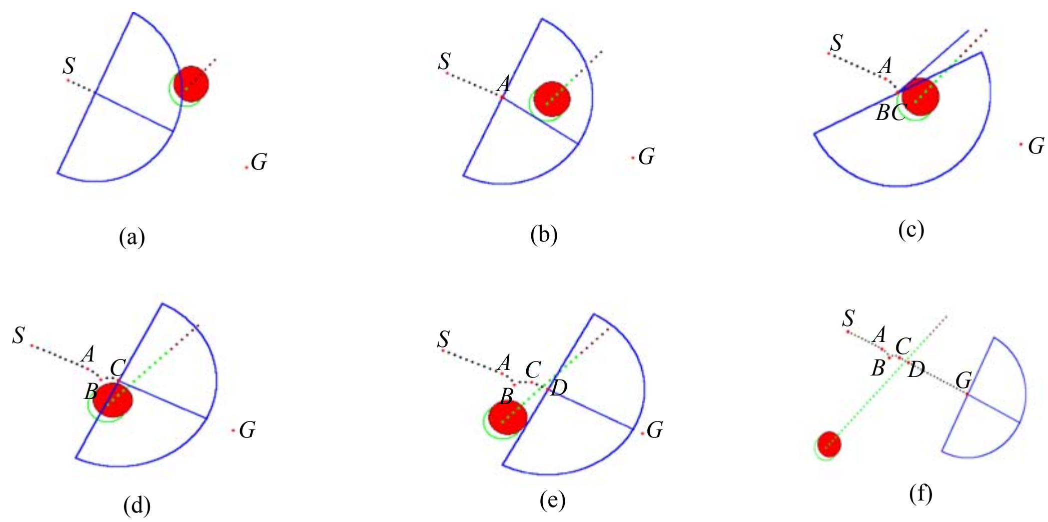

Fig.4 Robot detours dynamic obstacle in linear motion. (a) A dynamic obstacle comes into the robot's view; (b) The predicted position of the obstacle interfers the robot's motion to *G*; (c) The robot trys to detour the obstacle from its front; (d) The robot detours the obstacle form its back; (e) The obstacle disappears from the robot's view; (f) The robot moves straight to *G*

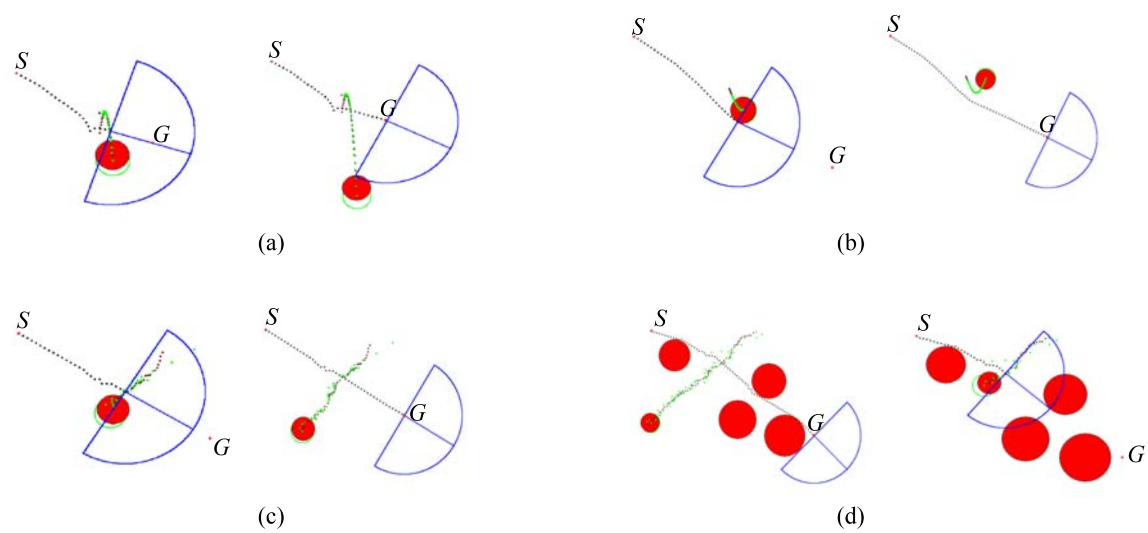

Fig.5 Robot detours dynamic obstacle in parabolic motion (a); in sinusoidal motion (b) and in stochastic motion (c); (d) On-line real-time path planning of mobile robot in dynamic uncertain environment

uncertain motion.

In Figs.4~5, red solid dot *S* and *G* respectively express the starting point and the goal point. Blue semicircle represents effective detection range of the robot's sensor system, that is, local planning window. Its black solid center stands for the mobile robot and the trajectory consisting of these centers represents the robot's motion path. The small red solid circle represents dynamic obstacle. Its brown solid center stands for real positions detected by the robot's sensor system and the trajectory consisting of these centers represents real trajectory of the dynamic obstacle. The small green hollow circle represents the dynamic obstacle on predicted position. Its green solid centers stand for the predicted positions computed by the AR model and the trajectory consisting of these centers represents predicted trajectory of the dynamic obstacle. And in Fig.8, the big red solid circle represents static obstacle.

Let us cite the case of dynamic obstacle in linear motion shown in Fig.4 to simply describe the planning process. Since no obstacle is detected in the robot's local planning window after the robot starts from the beginning point *S*, the robot moves straight to the goal point *G*. When the robot moves at the position shown in Fig.4a, a dynamic obstacle comes into the robot's vision. Therefore, sensors begin to sample the obstacle's real positions and its future positions are predicted by AR model described in Section 2. When the robot moves at the position *A* shown in Fig.4b, its sensors detect the moving obstacle in its way to the goal. Then desirable motion direction and local sub-goal of the robot are determined by the models and algorithm described in Section 3 above. As a result, the robot moves towards the edge of the dynamic obstacle gradually. As Fig.4c shows, position *B* is the inflexion of the robot's collision-free trajectory. Until the robot moves at the position *C* shown in Fig.4d, the robot escapes the moving obstacle and moves straight to the goal again. When the robot moves at the position *D* shown in Fig.4e, the moving obstacle disappears from the robot's vision. After position *C*, there is no obstacle detected in the way to the goal. So the robot will move straight until reaching the goal. The whole planning task is completed, as Fig.4f shows.

The above simulations show that this based on polar coordinates space approach to on-line real-time path planning of mobile robot is not only effective and feasible but also simple and flexible. The sizes and positions of static obstacles and the locations of starting point and goal point are all set up at random. Moreover, the velocity and motion direction of dynamic obstacle are all unknown. In addition, the case of dense obstacles is also considered, as Fig.8 shows. So the experiments have strong stochastic capability and reliability. Figs.4~5 show wonderful behavior of obstacle avoidance and perfect stability of the robot's motion because there are no phenomena of oscillation and hesitation shown in the above figures. Besides, the planning algorithm is so simple and fast that the robot can give a quick response when encountering obstacles.

CONCLUSION

Now on-line real-time path planning with obstacle avoidance for mobile robots in dynamic uncertain environment is a focus on robotics research. Previous methods of global planning and local planning have their respective drawbacks. Global planning methods can give optimal planning results. But they can only plan once off-line so that real-time capability cannot be executed. So their applications are limited. Though local planning methods can implement real-time planning, since there is no global information, such indices as motion path or runtime cannot be optimized so that it cannot meet the demands of the planning task.

To overcome their drawbacks, an on-line real-time path planning of mobile robots in dynamic uncertain environment is presented in this paper. In our approach, positions of moving obstacles are sampled by the robot's sensor system, and the sampling position information on dynamic obstacles is treated as instantaneously static. With current sampling positions, the AR model predicts future obstacle's positions in the next sampling duration. Then the robot's motion path is planned with such predicted positions. Thus, dynamic collision-free path planning is translated into static one.

With the integration of global planning and local planning, an on-line real-time path planning approach for mobile robots is proposed. This approach is based on polar coordinates in which the desirable direction angle is taken into consideration as an optimization index. Completely different from those traditional path planning methods based on configuration space and Cartesian coordinates space, this approach does not concern how long the robot moves but in what direction it moves, that is, determining desirable direction of the robot's motion is just the task in every planning window.

With environment information detected by sensors, the difference between real direction angle and desirable direction angle of the robot's motion, considered as a drive, navigates the mobile robot to avoid obstacles and advance towards the goal step by step. Moreover, the decision of desirable direction angle is very simple so as to guarantee real-time performance. Therefore, this approach overcomes global planning's drawback of on-line real-time planning. In each planning window, minimizing the angle (desirable direction angle) between desirable direction of the robot and polar axis direction of which current robot's position points towards the goal is taken into consideration as an optimization index to plan local path. Since such index includes the global information of the goal, it is guaranteed that the planned path is optimal or sub-optimal. Hence, desirable direction angle minimization as an optimization index indicates the optimization capability of this approach so that it can overcome local planning's drawback of path optimization.

References

Fujumura, K., 1992. Route Planning for Mobile Robots amidst Moving Obstacles. IROS'92, p.1751-1758.

Fiorinio, P., Shiller, Z., 1998. Motion planning in dynamic environments using velocity obstacles. *Int. Journal of Robotics Research*, **17**(7):760-772.

Kalman, R., 1960. A new approach to linear filtering and prediction problem. *Journal of Basic Engineering*, **82**(Series D):35-45.

Lee, D.T.L., Morf, M., Friedlander, B., 1981. Recursive least squares ladder estimation algorithms. *IEEE Trans. Acoustics, Speech, Signal Processing*, **29**(3):627-641. [doi:10.1109/TASSP.1981.1163587]

Luo, R., Chen, T., 1999. Target Tracking by Grey Prediction Theory and Look-ahead Fuzzy Logic Control. Proceedings of the IEEE International Conference on Robotics and Automation, p.1176-1181.

Makhoul, J., 1975. Linear Prediction: A Tutorial Review. Proc. IEEE, **63**:561-580.

Rude, M., 1997. Collision avoidance by using space-time representations of motion processes. *Autonomous Robots*, **4**(1):101-119. [doi:10.1023/A:1008863220740]

Sen, A., Srivastava, M., 1990. Regression Analysis: Theory, Methods and Applications. Springger-Verlay, New York.

Shensa, M.J., 1981. Recursive least squares lattice algorithms—A geometrical approach. *IEEE Trans. Auto. Control*, **26**(3):695-702. [doi:10.1109/TAC.1981.1102682]

Takeshi, A., 1994. Motion Planning for Multiple Obstacles Avoidance of Autonomous Mobile Robot Using Hierarchical Fuzzy Rule. Proceedings of IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent System (MFI'94), Las Vegas, p.265-271.

Tang, P., Zhang, Q., Yang, Y.M., 2000. Studying on Path Planning and Dynamic Obstacle Avoiding of Soccer Robot. Proceedings of the 3rd World Congress on Intelligent Control and Automation, University of Science and Technology of China, Hefei, p.1244-1247.

Zhu, Q., 1991. Hidden markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Trans. on Robotics and Automation*, **7**(3):390-396. [doi:10.1109/70.88149]