第29卷 第1期 自 动 化 学 报 Vol. 29, No. 1
2003年1月 ACTA AUTOMATICA SINICA Jan., 2003

Robot Rolling Path Planning Based on Locally Detected Information[1)]

ZHANG Chun-Gang XI Yu-Geng
(*Institute of Automation, Shanghai Jiaotong University, Shanghai* 200030)
(E-mail: cgzhang925@online.sh.cn; ygxi@sjtu.edu.cn)

Abstract Using the path planning method based on rolling windows, robot path planning in a globally unknown environment is studied. The method makes full use of real-time environmental information locally detected by the robot and the on-line planning is performed in a rolling style. Mechanisms of optimization and feedback are combined in a feasible way. The subgoal determination strategy of rolling path planning is analyzed according to various convex obstacle environment. And the accessibility of the planning algorithm is also discussed.

Key words Robot path planning, rolling planning, local planning, subgoal, accessibility

1 Introduction

Path planning is an important issue in robotics. In many cases, the environmental information for path planning is incomplete or uncertain. The robot can only detect local sensory information of the environment. People have proposed a few methods and strategies for path planning in unknown environments. But most of these methods have strict requirements on robots and can hardly guarantee the global convergence[1,4]. Borenstein[2] used VFF method to solve the collision-free problem in unknown environments. The method cannot guarantee that the robot reaches the goal successfully in some situations and may result in instability[3]. Lumelsky[4] studied nonheuristical methods of path planning in unknown environments. In his methods, the robot must remember some special points in the workspace and the planning is always guided by a global convergence criterion. However, the paths are not ideal in many cases because of lacking optimization. Iyengar[5] solved this problem by imparting the learning capability to the robot. The robot explores the obstacles using sensors and incrementally builds the terrain model. The computation burden of this method is heavy and the robot needs to store much information.

In fact, the path-planning problem of mobile robot in a globally unknown environment concerns how to make use of detected local information to plan a path that is not only feasible but also as optimal as possible. Both optimization and feedback should be considered in the planning. Using the rolling optimization concept in predictive control[6], [7] proposed a novel idea of path planning based on rolling windows. But in order to guarantee the accessibility of the planning, some strict constraints were bounded to the obstacle environment in [7]. This largely restricts the use of rolling path planning in real application. In this paper, path planning in an unknown environment with general convex obstacles is studied and an effective rolling path planning method is presented. The accessibility of the planning algorithm is discussed in details. The method has valuable meanings to real applications.

2 Problem formulation

Consider a mobile robot (Rob) in a two-dimensional unknown workspace (WS) of finite size. The robot is required to move autonomously from a start point (P_s) to a goal

1) Supported by the National 973 Plan (G1998030415), by the National 863 Program (2001AA422140) and by the National Natural Science Foundation of P. R. China (69934020)
Received June 27, 2001; in revised form April 29, 2002
收稿日期 2001-06-27; 收修改稿日期 2002-04-29

point (P_g) successfully in a finite time.

Rob has no priori knowledge of the workspace. At any instant, Rob can only scan a local circular region around itself, whose radius equals to r. WS is convex and is arbitrarily cluttered with finite number of static convex obstacles ($Obs_1, Obs_2, \cdots, Obs_n$). Rob is modeled as a point by "enlarging" the obstacle-size and for the requirement on safety. The boundaries of enlarged obstacles are safe regions that the point robot can move along. Obs_i ($i=1,2,\cdots,n$) is convex and with finite size. They do not intersect with each other or with the workspace boundary.

Set up the system Cartesian coordination in WS. Thus, $\forall P \in W$ has definite coordinates (x,y). At instant t, the position of Rob in WS is denoted as $P_R(t)$ and its coordinates are $(x_R(t), y_R(t))$. Let $t_1=0$ be the start instant of the planning.

Let $d(P_i, P_j)$ denote the distance from P_i to P_j, which is defined as:

$$d(P_i,P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

In the following discussion, we use superscripts "o" and "c" to represent the interior and the complement of the corresponding set, respectively.

Assume that all the points in WS make up a closed convex set W and its boundary is denoted as ∂W. All the points in $Obsi(i=1,2,\cdots,n)$ make up a closed convex set O_i and its boundary is denoted as ∂O_i. Then the feasible region[6] for Rob could be expressed as:

$$FD = W \cap (\bigcap_{i=1}^{n} (O_i^o)^c) \tag{2}$$

Definition 1. $T=[\tau_1, \tau_2]$, $\forall P_0 \in FD$, $\forall P_f \in FD$, if a continuous mapping FS: $T \rightarrow W(X \subseteq W)$ makes $FS(\tau_1)=P_0$, $FS(\tau_2)=P_f$, $FS(t) \in FD$, $t \in (\tau_1, \tau_2)$, then FS is called *a* feasible path from P_0 to P_f in X. The image set $FS(T)$ is called *a* passage from P_0 to P_f in X and denoted as $FP(P_0 P_f)$.

If there exists a feasible path form P_0 to P_f in X, then it is called that P_0 is connectible with P_f in X.

In the following, the rolling path planning method will be discussed in details.

3 Rolling path planning based on local information

Due to lack of enough environmental information, the robot cannot carry on one-off global optimization. It can only perform online path planning based on locally detected information. Instead of the one-off global optimization, the rolling path planning executes local planning repeatedly and makes full use of the newest local environmental information detected.

Definition 2. $Win(P_R(t_k)) = \{P \mid P \in W, d(P, P_R(t_k)) \leqslant r\}$ is called the vision scope of Rob at $P_R(t_k)$, namely its rolling window at $P_R(t_k)$, where $P_R(t_k) \in FD$ is the position of Rob at t_k (the beginning instant of the kth local planning).

The path planning algorithm based on rolling windows depends on real-time environmental information detected locally and the on-line path planning is performed in a rolling style. At each step of rolling planning, Rob generates an optimal subgoal based on the locally detected information by a heuristical method and plans a local path within the current rolling window. Then it moves a step along the local path. With the rolling window moving forward, Rob obtains the newer environmental information. Thus, optimization and feedback are combined in the rolling procedure. The general rolling path planning algorithm is presented as follows[6].

Algorithm 1. Path planning based on rolling windows.

Step1. If the goal is reached, the planning stops.

Step2. Update the environmental information in the current rolling window.

Step3. Generate a local optimal subgoal $P_{\text{sub}}(t)$.

Step4. Plan a proper local path in the current rolling window according to the subgoal and locally detected environmental information.

Step5. Move a step along the local path.

Step6. Return to Step1.

The subgoal generated in Step 3 is a mapping point of the global goal P_g in the current rolling window. It must reside in the feasible region and satisfy some optimal criterion. The determination of subgoals should not only make full use of the local information detected in real time, but also be consistent with the global goal. To maximize the use of the detected information, the subgoal should be chosen on the boundary of the detectable region.

Let $SW(t_k)=\{i \mid O_i^o \cap Win(P_R(t_k)) \neq \Phi\}$ be the subscript set of the obstacles detected at instant t_k in the rolling window. Let $\beta(t_k)=d(P_{sub}(t_k), P_g)$ be the distance from the local subgoal to the global goal at t_k.

Definition 3. The set of all points on $\partial Win(P_R(t_k))$ that are connectible with $P_R(t_k)$ within the rolling window $Win(P_R(t_k))$ is called the selectable set of subgoal at t_k, which is denoted as $\theta(t_k)$. Obviously, $\theta(t_k) \subseteq \partial Win(P_R(t_k)) \cap FD$.

The general subgoal determination method is described as follows.

1) If $P_g \in Win(P_R(t_k))$ and $P_R(t_k)$ is connectible with P_g in the rolling window, then $P_{sub}(t_k)=P_g$.

2) In other cases, $P_{sub}(t)$ could be determined by

$$\begin{aligned} &\min_P f(P) = g(P) + h(P) \\ &\text{s. t.} \quad P \in \partial\theta(t_k) \end{aligned} \tag{3}$$

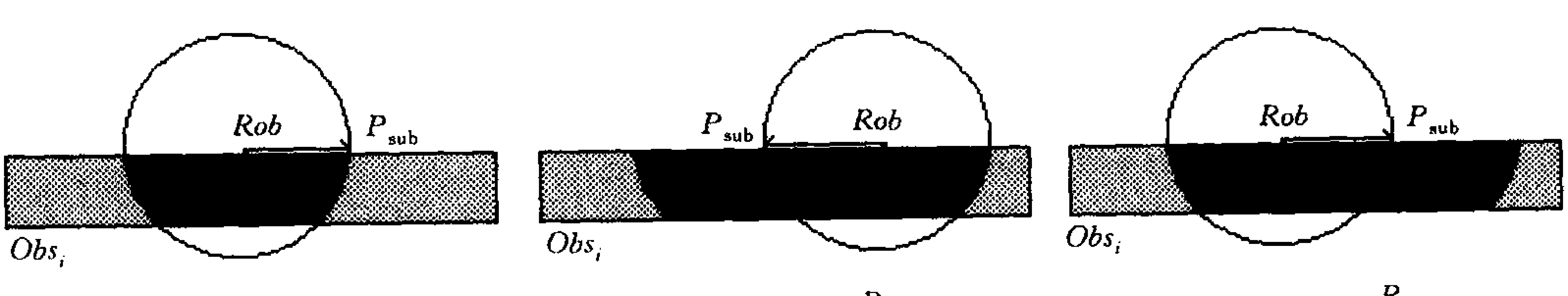
where $f(P)$ is a heuristical function. $g(P)$ denotes the cost from current position $P_R(t)$ to P, which can be estimated by the position of P and the detected environmental information in the current window. $h(P)$ denotes the cost from P to the goal point P_g. Since the environmental information out of the window is unknown, $h(P)$ is usually estimated by the distance from P to P_g. To reduce the computation burden, (3) can be transformed to the following optimization problem:

$$\min J = \min_{P_{sub}(t_k)} d(P_{sub}(t_k), P_g) \tag{4}$$

$$\text{s. t.} \quad P_{sub}(t_k) \in \theta(t_k)$$

It has been proved in [6] that the above method can guarantee the existence of subgoals and avoid deadlocks in some strictly constrained environment during the rolling planning. If there exist more than one subgoals, Rob can choose one arbitrarily.

Although the above method can guarantee the existence of subgoals in general obstacle environment, it might cause oscillations sometimes and Rob could not reach the global goal. For example, Rob chooses the connectible point $P_{sub}(t_{k-1})$ as its current subgoal based on the shortest distance from P_g at instant t_{k-1} (See Fig. 1(a)) and moves a step towards the subgoal. The new position is denoted as $P_R(t_k)$. At the instant t_k, the relative



(a) The $(k-1)$th planning (b) The kth planning (c) The $(k+1)$th planning

Fig. 1 Oscillation caused by improper determination of subgoals position of Rob and P_g has changed (See Fig. 1(b)). If Rob still determines its subgoal by (4), the new subgoal will again lead Rob to $P_R(t_{k-1})$ (See Fig. 1(c)). Thus, Rob will oscillate between $P_R(t_{k-1})$ and $P_R(t_k)$.

So in this case, the subgoal determination method should be modified. For further discussion, we give some related definitions first.

$\varphi(P_iP_j)$ denotes a directional beeline from P_i to P_j in WS. For simplicity, $\varphi(P_iP_j)$ also denotes the point set on that line.

If $\varphi(P_iP_j)\cap O_i^o\neq\Phi$, then $\varphi(P_iP_j)$ must intersect ∂O_i at two points, see Fig. 2.

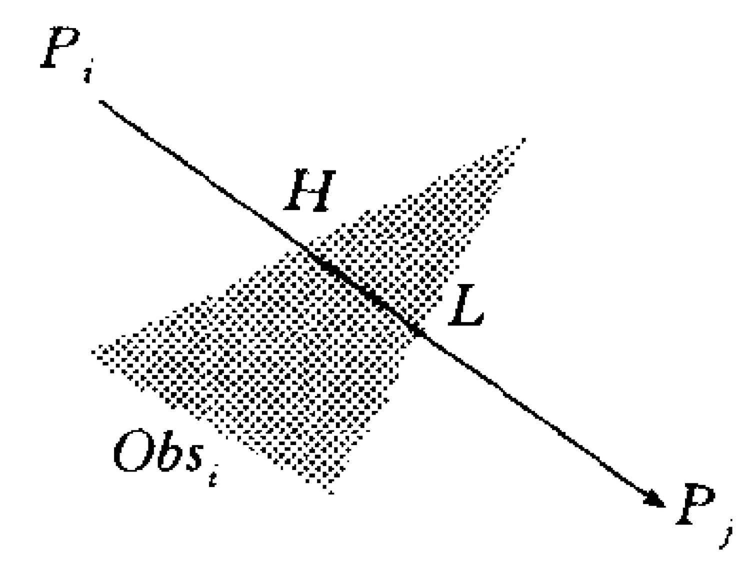
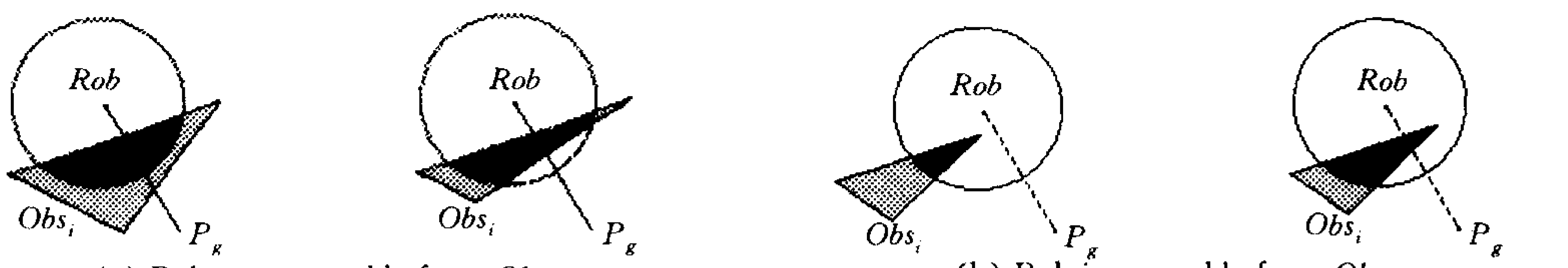


Fig. 2 H-hitting point; L-leaving point

Definition 4. If $\varphi(P_R(t_k)P_g)\cap O_i^o\neq\Phi$ ($i\in SW(t_k)$) and there is no leaving point L or $P_R(t_k)$ is not connectible with L within $Win(P_R(t_k))$, it is called that Rob is inescapable from Obs_i. Otherwise, it is called that Rob is escapable from Obs_i.

Fig. 3(a) and Fig. 3(b) show the two situations respectively.



(a) Rob is inescapable from Obs_i (b) Rob is escapable from Obs_i

Fig. 3 Relative positions between *Rob* and Obs_i

In some cases, Rob may be inescapable from several obstacles in its rolling window. In the following, we only refer to the nearest obstacle from Rob.

At instant t_k, if Rob is inescapable from Obs_i ($i\in SW(t_k)$) and $P_R(t_{k-1})\in\partial O_i$, $P_R(t_k)\in\partial O_i$, as in Fig. 1(a) and Fig. 1(b), then the subgoal should be determined as follows:

$$P_{sub}(t_k)\in\theta(t_k)\cap\partial O_i \tag{5a}$$

$$\text{s.t.}\quad FP(P_R(t_k)P_{sub}(t_k))\cap FP(P_R(t_{k-1})P_{sub}(t_{k-1}))=\Phi,\ \text{except for } P_R(t_k) \tag{5b}$$

where (5b) means that the new planned passage has no superposition with the last one. This indicates that the subgoal of the kth local planning should not be chosen as in Fig. 1 (b), which would result in oscillation. Instead, the subgoal should be the right intersection point of the boundary of the rolling window and the boundary of the obstacle. So local oscillations could be avoided and the robot won't repeat the traveled path if subgoals could be chosen properly by (4) and (5) according to various situations. The accessibility to the goal will be discussed in the next section.

When the subgoal has been determined, Rob plans a local optimal path based on the environmental information within the current rolling window and moves a step along it. For simplicity, we define a step for Rob to move from its current position to the subgoal after the kth local planning as $P_R(t_{k+1})=P_{sub}(t_k)$.

4 Analysis of accessibility

Using Algorithm 1, Rob performs rolling path planning in the unknown environment and moves towards the global goal. During the rolling planning process, Rob is guided by a series of subgoals and moves along local optimal paths bypassing all detected obstacles. Rob can be escapable from all obstacles gradually and moves along a strictly nearer path[6] to the global goal in the final phase.

In the following, we will prove that Rob can surely reach the global goal in a finite time in the unknown environment discussed.

Lemma 1. If $P_R(t_k)\in\partial O_i$ and $P_{\text{sub}}(t_k)\in\partial O_i$, then $FP(P_R(t_k)P_{\text{sub}}(t_k))\subset\partial O_i$. That is, if the current position and the subgoal are on the boundary of the same obstacle, then the passage of the current local optimal path must follow the obstacle boundary.

Since the obstacles are convex, it is easy to be proved.

Lemma 2. If Rob is escapable from $Obs_i(\forall i\in SW(t_k))$ at instant t_k, then there exists $\beta(t_{k-1})>\beta(t_k)$. Particularly, $\beta(t_k)=\beta(t_{k-1})-r$ when $P_g\notin Win(P_R(t_k))$.

Proof. If $P_g\in Win(P_R(t_k))$, then $P_{\text{sub}}(t_k)=P_g$, since $P_R(t_k)$ is connectible with P_g within the rolling window. Thus we have $\beta(t_k)=0<\beta(t_{k-1})$. If $P_g\notin Win(P_R(t_k))$, there exists $P_{\text{sub}}(t_k)\in\partial Win(P_R(t_k))\cap\varphi(P_R(t_k)P_g)$ according to the subgoal determination method. Because Rob is escapable from $Obs_i(\forall i\in SW(t_k))$ and $P_R(t_k)=P_{\text{sub}}(t_{k-1})$, we have $\beta(t_k)=\beta(t_{k-1})-r$. □

Lemma 2 shows that if Rob is escapable from $Obs_i(\forall i\in SW(t_k))$, its subgoals are strictly nearer to the global goal.

Theorem 1. If Rob is inescapable from Obs_i at instant t_k, it can surely turn to be escapable after finite steps.

Proof. If it could not turn to be escapable, then Rob would move along the obstacle boundary unilaterally without any repeat. That means the obstacle should be infinitely large, which contradicts the constraints on the environment. □

Lemma 3. If Rob turns to be escapable from Obs_i from original inescapable status, then it won't turn to be inescapable status again afterwards.

Since the obstacle is convex and with finite size, it is easily known to be true according to the subgoal determination method.

Theorem 2. There exists $t_D\geqslant 0$ such that $\beta(t_{k-1})>\beta(t_k)$ for $\forall t_k\geqslant t_D$; and there exists $t_F\geqslant t_D$ such that Rob is escapable from $Obs_i(\forall i\in SW(t_k))$ for $\forall t_k\geqslant t_F$.

Proof. If Rob is escapable from $Obs_i(\forall i\in SW(t_k))$, then there must exist $\beta(t_{k-1})>\beta(t_k)$ by Lemma 2. If Rob is inescapable from Obs_i, it can surely turn to be escapable after finite steps by Theorem 1. According to Lemma 3, it is an unreversed process that Rob turns to be escapable from its original inescapable status. Since there is finite number of obstacles in the environment, Rob can turn to be escapable from $Obs_i(\forall i\in SW(t_k))$ finally. So after enough times (denoted as D) of rolling planning, there will exist $\beta(t_{k-1})>\beta(t_k)$ for $\forall t_k\geqslant t_D$. Besides, there exists $\exists t_F\geqslant t_D$ such that Rob is escapable from $Obs_i(\forall i\in SW(t_k))$ for $\forall t_k\geqslant t_F$. □

Theorem 2 indicates that the subgoals will be strictly nearer to the global goal after finite times of rolling planning.

Theorem 3. Rob can surely reach the global goal in a finite time in the constrained environment.

Proof. By Theorem 2, we know that Rob is escapable from $Obs_i(\forall i\in SW(t_k))$ for $\forall t_k\geqslant t_F$. Then the subgoals will be strictly nearer to the global goal afterwards. Thus there exists $t_w\geqslant t_F$ such that $P_g\in Win(P_R(t_w))$ and at that time $P_R(t_w)$ is connectible with P_g within the rolling window. Then the subgoal is determined as $P_{\text{sub}}(t_w)=P_g$. And because $P_R(t_{w+1})=P_{\text{sub}}(t_w)$, Rob could surely reach the global goal in a finite time. □

5 Simulation results

Fig. 4 shows a simulation process of the robot rolling path planning in a globally unknown environment with convex obstacles. The mobile robot has no priori knowledge of the environment and can only scan a local circular region around itself. Using real-time information detected in its local window, the robot performs rolling path planning in the unknown environment. The unknown obstacles are marked with gray color and the detected ones with black color in the figure. The robot can move successfully from the start point (S) to the goal point (G). Since each planning is inside the local window, rolling path planning is fast and efficient.

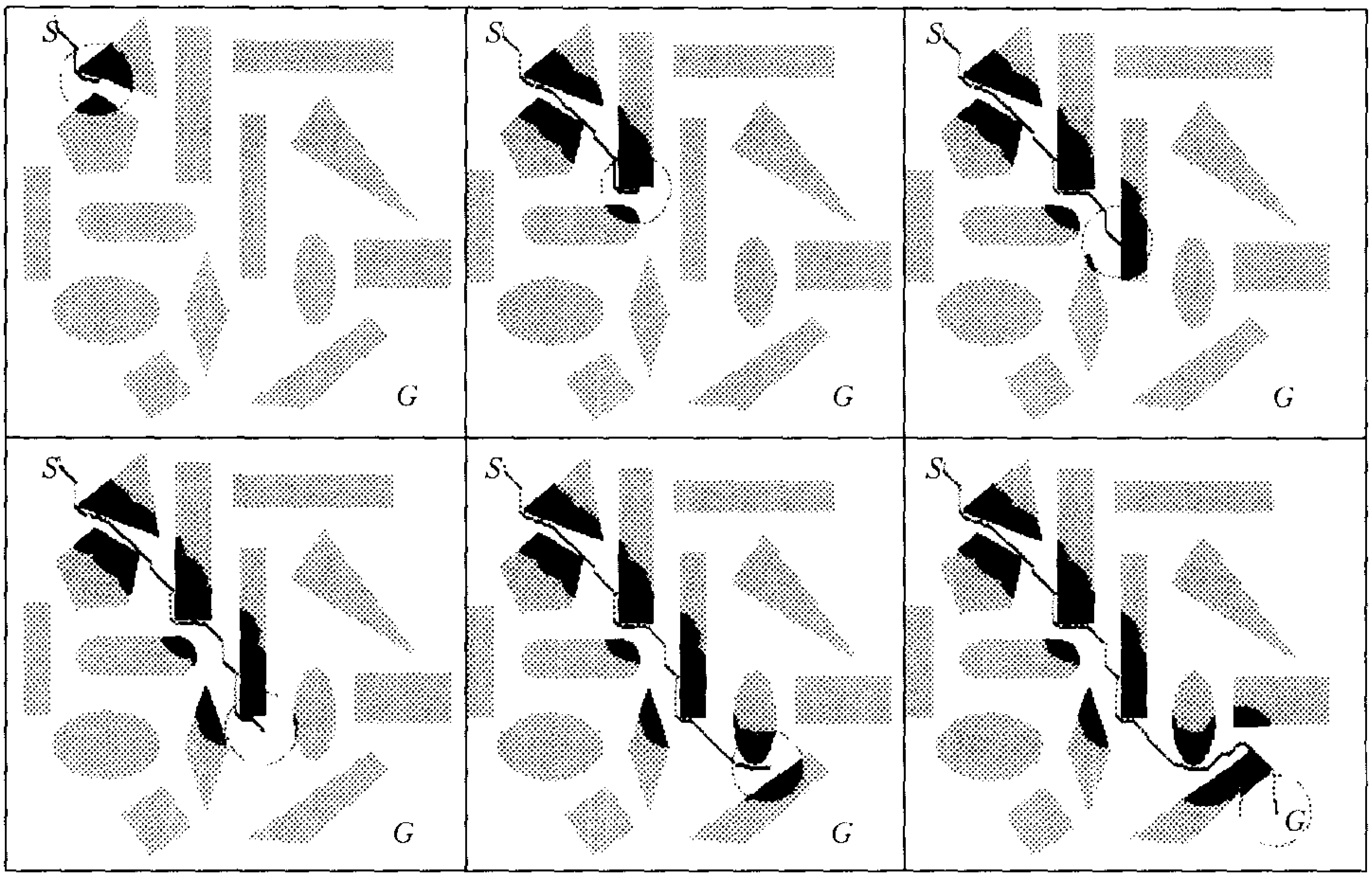


Fig. 4 Mobile robot rolling path planning in the unknown environment with convex obstacles

6 Conclusions

The problem of mobile robot path planning in an unknown environment is a key issue in robotics. Off-line global planning methods are prone to fail due to lack of priori environmental information. Furthermore, the robots are always constrained by finite sensorial scope in real applications. Only locally detected information could be used during the planning procedure. The path planning method based on rolling windows uses the principles of predictive control and combines the mechanisms of optimization and feedback. It performs in a rolling style and could solve the planning problem in an uncertain environment efficiently with low calculation burden. The accessibility of the algorithm is also guaranteed during the rolling planning process.

References

1 Sankaranarayanan A, Vidyasagar M. A new path planning algorithm for moving a point object amidst unknown obstacles in a plane. In: Proceedings of IEEE Conference on Robotics and Automation, France: Nice, 1990. 1930～1936

2 Borenstein J, Koren Y. Real time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 1989, **19**(5):1179～1187

3 Tilove R B. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In: Proceedings of IEEE Conference on Robotics and Automation, France: Nice, 1990. 566～571

4 Lumelsky V J. Algorithm and complexity issues of robot motion in an uncertain environment. *Journal of Complexity*, 1987, **3**(2):146～182

5 Iyengar S S, Jorgensen C C, Rao S V N, Weisbin C R. Learned navigation paths for a robot in unexplored terrain. In: Proceedings of 2nd Conference on Artificial Intelligence Applications and Engineering of Knowledge Based Systems, USA: Miami Beach, Florida, 1985. 11～13

6 Xi Yu-Geng. Predictive control. Beijing: National Defense Industry Press, 1993(in Chinese)

7 Zhang Chun-Gang, Xi Yu-Geng. Robot path planning in globally unknown environments based on rolling windows. *Science in China*(*E*), 2001, **44**(2): 131～139(in Chinese)

ZHANG Chun-Gang Ph. D. candidate of Automation Department, Shanghai Jiaotong University. His research interests include robot path planning and coordination of multi-robots.

Xi Yu-Geng Professor of Shanghai Jiaotong University. Received his Ph. D. degree from Technical University of Munich, Germany in 1984. His research interests include predictive control, large-scale system and intelligent robotics.

基于局部探测信息的机器人滚动路径规划

张纯刚　　席裕庚

(上海交通大学自动化研究所　上海　200030)

(E-mail: cgzhang925@online. sh. cn; ygxi@sjtu. edu. cn)

摘　要　用基于滚动窗口的路径规划方法研究了全局环境未知时的机器人路径规划问题.该法充分利用机器人实时测得的局部环境信息,以滚动方式进行在线规划,实现了优化与反馈的合理结合.文中分析了不同凸障碍环境下滚动路径规划子目标选择策略,并且还探讨了规划算法的可达性.

关键词　机器人路径规划,滚动规划,局部规划,子目标,可达性

中图分类号　TP24