Robotica (2013) volume 31, pp. 1263–1274. © Cambridge University Press 2013
doi:10.1017/S0263574713000477

# Artificial moment method using attractive points for the local path planning of a single robot in complicated dynamic environments

Wang-bao Xu†,‡, Jie Zhao§, Xue-bo Chen‡* and Ying Zhang‡

†*School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, P. R. China*
‡*School of Electronics and Information Engineering, Liaoning University of Science and Technology, Anshan 114051, P. R. China*
§*State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, 150001, P. R. China*

(Accepted May 6, 2013. First published online: June 7, 2013)

**SUMMARY**

A novel path planner is presented for the local path planning of a single robot (represented with *R*) in a complicated dynamic environment. Here a series of *attractive points* are computed based on *attractive segments* for guiding *R* to move along a shorter path. Each attractive segment is obtained by using the full environmental knowledge and will be used for several sampling times in general. A motion controller, which is designed based on artificial moments and a robot model that has a principal motion direction line(PMDline), makes *R* move closely to attractive points while away from obstacles. Attractive and repulsive moments are designed, which only make *R*'s PMDline face toward attractive points and opposite to obstacles in general, as in most cases, *R* will move along its PMDline with its full speed. Because of the guidance of attractive points and *R*'s full-speed motion, the global convergence is guaranteed. Simulations indicate that the proposed path planner meets the requirements of real-time property while can optimize *R*'s traveling path.

KEYWORDS: Path planning; Autonomous robots; Motion control; Artificial moment method.

## 1. Introduction

According to refs. [1] and [2], path planning is an important problem in robotics. Given the geometry of a robot and obstacles, a planner is required to generate a collision-free path between an initial and a goal position, or find out that no such path exists. The basic path-planning problem has several classifications. A common one includes global and local planning.[3] In global or off-line planning, a robot knows the whole environment previously and plans the entire path *a priori* to its movement. In local or on-line planning, however, the environmental information has to be obtained by the sensors of the robot in real time as the environment is *a priori* unknown or partially unknown.

In reality, a robot cannot often know the whole environment previously. Therefore, much attention has been paid to the local path planning of mobile robots in unknown, complicated and dynamic environments,[4] and this paper also focuses on this problem.

The methods based on the visibility graph,[2,5,6] the Voronoi diagram,[2,5,7,8] and the tangent graph[9,10] are often used for the local path planning problems. The main idea of the methods consists of capturing the connectivity of the free space into a network of unidimensional curves called roadmap. Once the roadmap has been built, it is used as a set of standardized path, and the path planning is reduced to find a path between an initial and a final point on the roadmap.

The probabilistic roadmaps[1,11] and rapidly exploring random trees[1,12,13] are also frequently used for the local path planning problems. They are used in configuration spaces but have no requirement to construct explicit representations of C-space obstacles. The probabilistic roadmap planner uses random sampling to construct a probabilistic roadmap in the configuration space and then tries to find a path between any two input configurations by connecting them to the roadmap. The rapidly exploring random tree planner,[1,12,13] however, attempts to grow a tree from the initial configuration to the goal configuration by random sampling.

The above methods are of "global convergence" and their solutions are often optimal. However, they are hard to use in real time in local path planning as in these methods, a complete feasible path from the initial position to the goal position has to be computed, which often takes a great deal of computation. On the other hand, in local path planning, the environmental knowledge is changed frequently, and sometimes a small change may cause re-planning of almost a complete path. Therefore, these methods suffer from heavy computational burdens.

In contrast, the artificial potential field (APF) methods[14–19] which use local sensory information in a purely reactive fashion, are usually much simpler to be implemented. Here a repulsive potential field to force the robot away from obstacles and an attractive potential field to drive the robot close to a target are employed to generate a force. The force equals the negative gradient of the total potentials and makes the robot move from the position with the higher potentials to that with the lower. This method often suffers from the "local minimum" problem [15–19] and yields low quality solutions in terms of path length.

* Corresponding author. E-mail: xuebochen@126.com

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

The wall following or "bug" algorithms[4,19,20] work also in a reactive fashion. Two reactive modes of motion—moving toward the goal and following an obstacle boundary—are used in these algorithms. In order to guarantee convergence to the goal, the transition between the two modes is governed by a criterion to ensure that the distance to the goal decreases monotonously. These algorithms combine local planning with global information, and can guarantee the "global convergence" to a certain extent. However, they also often yield low quality solutions.

In fact, the path planning is an non-deterministic polynomial (NP)-complete problem,[2,21] and the global optimal solution is difficult to be obtained in many cases. Using less computation to obtain a shorter path is an accepted tradeoff. As such, the rolling window method[22–25] is developed. Here an optimal subgoal and an optimal local path to reach the subgoal are obtained within the current rolling window firstly at each step. The robot moves a step along the local path, and then the above steps are repeated until the task is fulfilled. The main advantage of this method is that it can often obtain a suboptimal path[25] with low computation cost as an optimal path within a rolling window is easier to be obtained. But in complicated environments, the algorithms might probably fail[24] as the local detected environmental information used by the robot is not enough to plan a globally feasible path in the complicated obstacle environment.

In this paper, a realistic path planner is presented for the local planning of a single robot in complicated, dynamic environments. Here a series of *attractive points* are used for guiding the robot to move along a shorter path. Different from the subgoals in the rolling window algorithms and the instant goals in ref. [4], which are computed based on local environmental knowledge, attractive points are computed based on *attractive segments*. Each attractive segment is obtained by using the full environmental knowledge, and will be used for several sampling times in general. A motion controller, which is designed based on artificial moments and a robot model that has a *principal motion direction line* (PMDline), makes the robot move closely to attractive points while away from obstacles. In most cases, the controller will drive the robot move along its PMDline with its longest step length. Therefore, attractive moments are designed only to make the robot's PMDline face toward attractive points and repulsive ones are only to make the robot's PMDline opposite to obstacles in general. Obviously, the artificial moment functions are different from the traditional artificial potential functions, which are designed to adjust the position and moving speed of the robot directly. As the robot always moves under the guidance of attractive points with its full speed, the robot is difficult to be trapped even in complicated environment, and then the global convergence is guaranteed.

The rest of this paper is organized as follows. In Section 2, associated models and concepts are introduced and the discussed problem is formulated. In Section 3, the algorithms for attractive segments, attractive points and the local path planning of a single robot are presented. In Section 4, a motion controller based on artificial moments is proposed. Simulations are given and analyzed in Section 5, and some conclusions are deduced in Section 6.

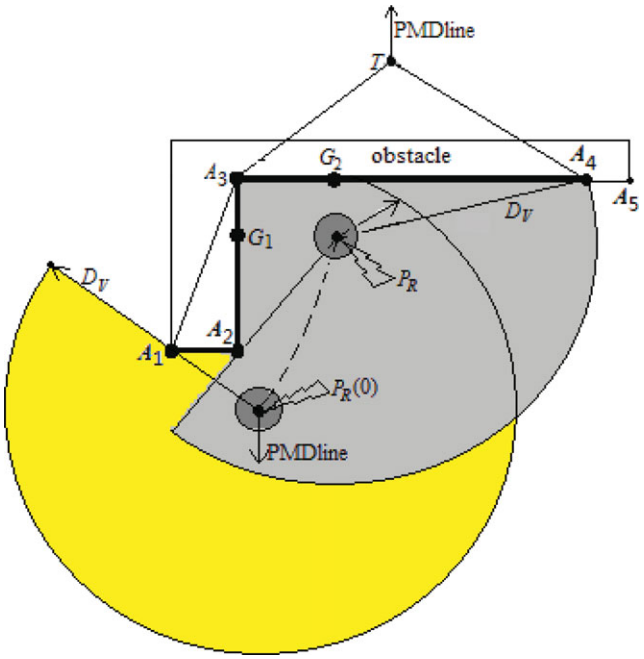



Fig. 1. (Colour online) Associated models.

## 2. Problem Formulation and Basic Definitions

This paper focuses on the path planning of a single robot in a bounded planar environment, which is populated with polygonal obstacles, and some obstacles may be in motion.

The robot model is a circle, as shown in Fig. 1, which has a PMDline as that in refs. [26] and [27]. The PMDline of the robot is a ray starting from its position (center) and can indicate its posture or motion direction.

The target model is a point with a PMDline, as shown in Fig. 1, which may also be in motion along its PMDline. When the target is static, its PMDline indicates its posture.

**Definition 1.** *Rotate a directed line (ray, directed segment) around its start point until its direction is same as that of the X-axis of the global coordinate system. As such, the angle formed by the above rotation is the direction angle of the line if the angle's absolute value is not larger than* $\pi$. *Assume that* $\beta_i$ *and* $\beta_j$ *are the direction angles of directed lines* $l_i$ *and* $l_j$, *and the function*

$$\mathrm{agl}(x) = \begin{cases} x & |x| \le \pi \\ x - 2\pi \ \mathrm{sign}(x) & |x| > \pi \end{cases}. \tag{1}$$

Then, $\beta = \mathrm{agl}(\beta_i - \beta_j)$ is the angle from $l_i$ to $l_j$.

For convenience, some notations which will be used throughout this paper are listed as follows:

$S_O$: longest step length of dynamical obstacles;
$R$, $T$: robot and target;
$\beta_T$, $S_T$: PMDline direction angle and longest step length of $T$;
$(x_T, y_T)^T$: coordinates of $T$'s position;
$\beta_R$, $S_R$: PMDline direction angle and longest step length of $R$;
$P_R$: $R$'s position (center);

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

$(x_R, y_R)^T$: coordinates of $P_R$;
$D_R$, $D_V$: radius of $R$ and valid radius of $R$'s sensor;
$D_S$: an important constant associated with $R$'s safety;
$A_1A_2$: line segment with end points $A_1$ and $A_2$;
$DS(A_1, A_2)$: directed line segment from point $A_1$ to point $A_2$;
$\beta(A_1, A_2)$: direction angle of $DS(A_1, A_2)$;
$RL(A_1, A_2)$: ray starting from $A_1$ and passing through $A_2$;
$\partial(W, -A_1, \ldots, -A_n)$: points on continuous curve $W$ except for points $A_1, \ldots, A_n$;
$|W|$: length of continuous curve $W$;
$\partial(RL(A_1, A_2), -A_1A_2)$: points on $RL(A_1, A_2)$ except for that on $A_1A_2$;
$A_1A_2, \ldots, A_n$: polyline with vertices $A_1, A_2, \ldots, A_n$.

For an arbitrary variable $V$, $V(k)$ denotes its value at sampling time $t_k$ and $V$ denotes it value at the current time.

$R$ has no *a priori* knowledge of the environment and, therefore, it is necessary for $R$ to continuously sense its surrounding environment using its onboard sensors. $R$'s sensor is omnidirectional. For a point $F$, only when $P_RF$ does not pass through any obstacle and $|P_RF| \leq D_V$ can $F$ be detected.

For example, in Fig. 1, only the points within the region with gray color or on the boundaries of the region can be detected at $P_R$. $A_1$ can be detected at $P_R(0)$ but cannot at $P_R$.

**Assumption 1.** *At each instant, R knows the position, step length and PMDline's direction of T through communication or through other means.*

**Assumption 2.** *$S_R$ is larger than $S_T$ and $S_O$; $D_S$ is larger than $2(S_O + S_R + D_R)$.*

**Remark 1.** *For R and an obstacle, if the shortest distance between their boundaries is not less than $D_S$, then the condition $2(S_O + S_R + D_R) < D_S$ can guarantee that no collision between them occurs even if they move directly toward each other with step length $S_O$, $S_R$, respectively.*

Once the models, assumptions, sensing capabilities highlighted above have been detailed, the problem reduces to find out a path planner that will enable $R$ avoid potential collisions with any present obstacles, reach $T$ or find out no such path exists in finite time. Furthermore, the path planner in this paper is required to make $|\beta_R(k) - \beta_T(k)|$ and $R$'s traveling path be as small/short as possible if $R$ reaches $T$ at time $t_k$ so that $R$ can have given posture and $R$'s traveling path can be optimized.

For further discussion, the rest of this section gives some important definitions which will be used in the following sections.

**Definition 2.** *Assume that $\Sigma$ is a polyline on the boundary of an obstacle. If all the points on $\Sigma$ have been detected, and for any continuous curve $\Sigma*$ such that $\Sigma*$ is on the same obstacle as $\Sigma$ and $\Sigma \subset \Sigma*$, $\Sigma*$ has at least one point which has never been detected, then $\Sigma$ is a knowledge obstacle wall.*

*For example, in Fig. 1, the polyline $A_1A_2A_3A_4$ is a knowledge obstacle wall, but $A_1A_2A_3A_4A_5$ is not because some points on it, like $A_5$, have never been detected.*

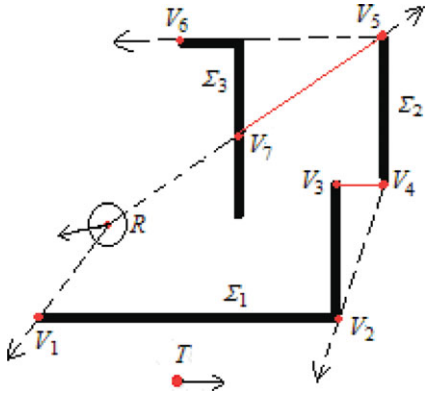



Fig. 2. (Colour online) Open tangent points and artificial obstacle segments.

*All knowledge obstacle walls will be memorized and updated in time.*

*All line segments on knowledge obstacle walls are called as knowledge obstacle segments.*

**Definition 3.** *For a point G on a knowledge obstacle wall $\Sigma$, if G can be detected at present, $|P_RG| \leq D_S$, and a constant $\delta > 0$ exists such that for any point A on $\Sigma$, $|AG| \leq \delta$ implies $|P_RA| \geq |P_RG|$, then G is a dangerous wall representative point.*

For example, $G_1$, $G_2$ in Fig. 1 are two dangerous wall representative points.

Obviously, a dangerous wall representative point is the point with the shortest distance from $P_R$ in a local region of a knowledge obstacle wall. As such, several such points may be on a same wall, which helps to prevent $R$ from colliding with one side while avoiding another side of the wall as each dangerous wall representative point will, and only such points can, generate a repulsive moment by which $R$ is forced away from the point.

**Definition 4.** *Assume that $F_1$, $F_2$ are two arbitrary points on a continuous curve W, and that S is a set such that its members are all continuous curves with ends $F_1$, $F_2$ and all points on its members are on or very close to W. For a knowledge obstacle wall $\Sigma$, if for any curve $W^*$ in S, $\partial(W*, -F_1, -F_2)$ shares at least one point with $\Sigma$, then W is said to be blocked by $\Sigma$.*

For example, in Fig. 1, $P_RA_2A_3T$ is blocked by $A_1A_2A_3A_4$, but $P_RA_4T$ and $P_RA_2A_1A_3T$ are not.

If $P_RT$ is blocked by only one knowledge obstacle wall, then the wall is denoted by $\Sigma_{\text{bw}}$. For example, $\Sigma_1$ in Fig. 2 is $\Sigma_{\text{bw}}$.

**Definition 5.** *Assume that $A_1$ is an end of a segment on a knowledge obstacle wall $\Sigma$, and that $P_1$ is a point different from $A_1$.*

1. If $RL(P_1, A_1)$ is not blocked by $\Sigma$, then $A_1$ is an *open tangent point* from $P_1$ to $\Sigma$.
2. When the set $\partial(RL(P_1, A_1), -P_1A_1) \cap \Sigma$ is not empty, assume that $A_2$ is the point with the shortest distance

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

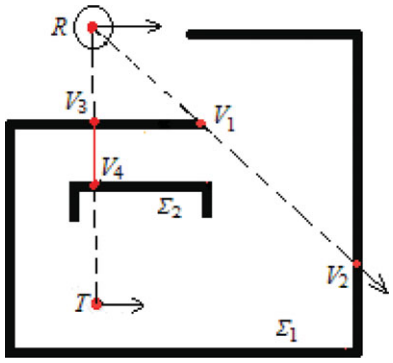



Fig. 3. (Colour online) Close tangent points and artificial obstacle segments.

from $A_1$ in the points such that $P_1A_2$ is not blocked by $\Sigma$. If a closed curve by which $P_1$ and $T$ are separated can be formed by $A_1A_2$ and a part of $\Sigma$, then $A_1$ is a *close tangent point* from $P_1$ to $\Sigma$.

For example, in Fig. 2, $V_5$ is an open tangent point from $P_R$ to $\Sigma_2$; $V_6$ is an open tangent point from $V_5$ to $\Sigma_3$. In Fig. 3, $V_1$ is a close tangent point from $P_R$ to $\Sigma_1$.

## 3. Important Algorithms for the Proposed Path Planner

### *3.1. Rules for setting artificial obstacle segments*

*Artificial obstacle segments* play an important role in the proposed path planner as if an artificial obstacle segment shares points with two or more knowledge obstacle walls, then these walls will be merged into one wall. For this reason, artificial obstacle segments can prevent $R$ from walking into some ways that are not suitable for $R$ to pass through and can decrease the computation for attractive segments.

There are two types of artificial obstacle segments, that is, *Narrow-type* ($N$-type) and *Back-type* ($B$-type).

The rules for setting $N$-types are as follows. For two knowledge obstacle segments $l_1$ and $l_2$ that have no shared point with each other, assume that $V_1$ is the point on $l_1$ with the shortest distance from $l_2$ and $V_2$ is the point on $l_2$ with the shortest distance from $V_1$. If $l_1$ has no shared points with the knowledge obstacle segments and artificial obstacle segments that have shared points with $l_2$, and $|V_1V_2| < D_S$, then $V_1V_2$ will be set as an $N$-type artificial obstacle segment.

$N$-type artificial obstacle segments are mainly used to prevent $R$ from walking into ways narrower than $D_S$ as $R$ cannot pass through them safely. For example, in Fig. 2, $V_3V_4$ is such one.

The rules for $B$-type artificial obstacle segments are as follows. Assume that $\Sigma_1$ is a knowledge obstacle wall blocking $P_RT$ and $\Sigma_2$ is a knowledge obstacle wall different from $\Sigma_1$. (1) If $\Sigma_2$ also blocks $P_RT$, then $A_1A_2$ can be set as a $B$-type artificial obstacle segment, where $A_1$ and $A_2$ are the intersections of $P_RT$ with $\Sigma_1$ and $\Sigma_2$, respectively. (2) Assume that $A_1$ is an open or close tangent point from $P_R$ to $\Sigma_1$. If $P_RA_1$ is blocked by $\Sigma_2$, then $A_1A_2$ can be set as a $B$-type artificial obstacle segment, where $A_2$ is an intersection of $P_RA_1$ with $\Sigma_2$.

For example, both $V_7V_5$ in Fig. 2 and $V_3V_4$ in Fig. 3 are $B$-type artificial obstacle segments.

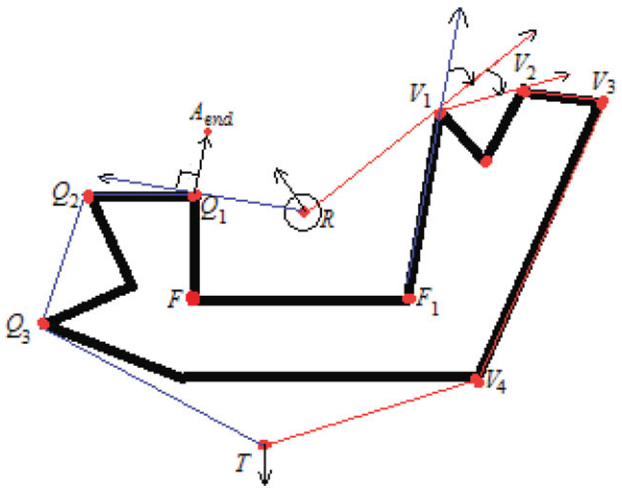



Fig. 4. (Colour online) Attractive segments and $FW(P_R \sim V_1 \sim T)$.

$B$-type artificial obstacle segments are mainly used to decrease the total number of knowledge obstacle walls by merging some different walls into one wall.

**Conclusion 1.** If no artificial obstacle segment can be set, then (1) $P_RT$ is blocked by at most one knowledge obstacle wall; (2) assume that $A_1$ is a close or open tangent point from $P_R$ to $\Sigma_{\text{bw}}$, and then $P_RA_1$ will not be blocked by knowledge obstacle walls.

The differences between artificial obstacle segments and knowledge obstacle segments are as follows:

1. When dangerous wall representative points are computed, artificial obstacle segments will be ignored, which implies that artificial obstacle segments will not impose repulsive moments on $R$.
2. Some artificial obstacle segments will be removed in the following rules.

   I. If a knowledge obstacle segment is found to be in motion, then all artificial obstacle segments associated with the segment will be removed.
   II. If a closed curve by which $R$ and $T$ are separated has been formed by a part of a knowledge obstacle wall, then all $B$-type artificial obstacle segments on the wall will be removed.

### *3.2. Algorithm for attractive segments*

An attractive segment is a directed segment whose starting point is denoted by $A_{\text{start}}$ and ending point is $A_{\text{end}}$.

For example, in Fig. 4, $DS(Q_1, A_{\text{end}})$ is an attractive segment.

When $P_RT$ is blocked by knowledge obstacle walls, an attractive segment is required to be obtained as one of its ends will be selected as the attractive point. A length value $A_{\text{length}}$ is also needed to be obtained as it will be used to determine whether the attractive segment at the next time $t_{k+1}$ can be same as that at the current time.

Once an attractive segment is obtained, it is regarded as a short-term goal for $R$ to pass through. For this reason, if no condition for changing attractive segments is satisfied, then the attractive segment will be same as that at the last time

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

$t_{k-1}$ and $R$ will still move toward the attractive point on the segment.

Let $S_{v1}$ denote the set of the visible knowledge obstacle segments (some points on them can be detected at current time) on the same knowledge obstacle wall as $A_{\text{start}}(k-1)$ and the artificial obstacle segments sharing points with the segments in $S_{v1}$. $S_{v2}$denotes the set of the visible knowledge obstacle segments that are not in $S_{v1}$ and the artificial obstacle segments sharing points with the segments in $S_{v2}$. $P_R A_{\text{end}}(k-1) + A_{\text{length}}(k-1)$ denotes the directed segment starting from $P_R$, passing through $A_{\text{end}}(k-1)$ and with length $|P_R A_{\text{end}}(k-1)| + A_{\text{length}}(k-1)$.

Then, the conditions for changing attractive segments are given as follows:

(1) $P_R(k-2)P_R$has an intersection with $RL(A_{\text{start}}(k-1), A_{\text{end}}(k-1))$. This condition implies that $R$ has passed through $RL(A_{\text{start}}(k-1), A_{\text{end}}(k-1))$. Thus, a new different attractive segment should be obtained.
(2) $P_R A_{\text{end}}(k-1) + A_{\text{length}}(k-1)$ has an intersection with the segments in $S_{v1}$.
(3) Both $P_R A_{\text{end}}(k-1)$ and $P_R A_{\text{start}}(k-1)$ have been blocked by the segments in $S_{v2}$.

Conditions (2) and (3) imply that the environment has changed considerably. Then a new attractive segment should be obtained.

In fact, the conditions for changing attractive segments are difficult to be satisfied. As such, an attractive segment can be used for several sampling times in general.

When the attractive segment at current time cannot be same as that at $t_{k-1}$, the following conclusion and algorithm are useful for obtaining it.

**Conclusion 2.** If $V_1$ is an open or close tangent point from $P_R$ to $\Sigma_{\text{bw}}$, and $T$ is reachable, then a polyline $P_R V_1 V_2, \ldots, V_n T$ denoted by $FW(P_R \sim V_1 \sim T)$ can be obtained by Algorithm 1:

**Algorithm 1.** Computing $FW(P_R \sim V_1 \sim T)$.

1. Let $V_0 = P_R$, $V_i = V_1$, and then go to step 2.
2. If $V_i T$ is not blocked by $\Sigma_{\text{bw}}$, then stop the algorithm and output $FW(P_R \sim V_1 \sim T)$. Otherwise, go to step 3.
3. If there is a close tangent point from $V_i$ to $\Sigma_{\text{bw}}$, then the point is $V_{i+1}$. Let $V_i = V_{i+1}$, and then return to step 2. Otherwise, go to step 4.
4. Obtain a point $F_i$ such that $F_i$ is different from $V_{i-1}$ and $V_i$, and $F_i$, $V_i$ are two ends of a same knowledge obstacle segment.
5. $V_{i+1}$ is the open tangent point from $V_i$ to $\Sigma_{\text{bw}}$ such that $\text{sign}(\text{agl}(\beta(V_{i-1}, V_i) - \beta(V_i, V_{i+1})))$ is same as $\text{sign}(\text{agl}(\beta(F_i, V_i) - \beta(V_{i-1}, V_i)))$, as shown by $V_0$, $V_1$, $V_2$ and $F_1$ in Fig. 4. Let $V_i = V_{i+1}$, and then return to step 2.

For example, in Fig. 4, $FW(P_R \sim V_1 \sim T)$ is $P_R V_1 V_2 V_3 V_4 T$ and $FW(P_R \sim Q_1 \sim T)$ is $P_R Q_1 Q_2 Q_3 T$.

Based on Conclusion 2, Algorithm 2 is presented for obtaining the attractive segment at current time when it cannot be same as that at $t_{k-1}$ and $P_R T$ is blocked by knowledge obstacle walls.

**Algorithm 2.** Obtaining $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ when they cannot be same as that at $t_{k-1}$.

1. If a closed curve by which $R$ and $T$ are separated is formed by a part of a knowledge obstacle wall, then remove all $B$-type artificial obstacle segments on the wall. If such a closed curve still exists, then $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ are empty which imply that $T$ is unreachable; stop the algorithm.
2. Set $B$-type artificial obstacle segments until no such segments can be set.
3. If there is a close tangent point from $P_R$ and $\Sigma_{\text{bw}}$, then the point is taken as $A_{\text{start}}$; go to step 6. Otherwise, go to step 4.
4. Obtain $FW(P_R \sim V_1 \sim T)$ and $FW(P_R \sim Q_1 \sim T)$ by Algorithm 1, where $V_1$, $Q_1$ are the two open tangent points from $P_R to \Sigma_{\text{bw}}$.
5. If $|FW(P_R \sim V_1 \sim T)| < |FW(P_R \sim Q_1 \sim T)|$, then $V_1$ is taken as $A_{\text{start}}$; otherwise, $Q_1$ is $A_{\text{start}}$.
6. Assume that $F$, $A_{\text{start}}$ are two ends of a knowledge obstacle segment. Then $A_{\text{end}}$ is the point such that $\text{agl}(\beta(A_{\text{start}}, A_{\text{end}}) - \beta(P_R, A_{start})) = \text{sign}(\text{agl}(\beta(P_R, A_{\text{start}}) - \beta(P_R, F)))\pi/2$ and $|A_{\text{start}} A_{\text{end}}| = D_S/2$, as shown in Fig. 4.
7. If $RL(P_R, P_{\text{end}})$ has no intersection with $\Sigma_{\text{bw}}$, then $A_{\text{length}} = D_V$. Otherwise, assume that $A$ is the intersection with the shortest distance from $P_{\text{end}}$ in the intersections, and then $A_{\text{length}} = |P_{\text{end}} A|/2$.

Note that if $A_{\text{start}}$ is on a moving knowledge obstacle segment, then $A_{\text{start}}$ and $A_{\text{end}}$ will move synchronously with the segment.

### *3.3. Algorithm for attractive points*

Based on Algorithm 2, Algorithm 3 is presented for the attractive point $P_{\text{att}}$ at current time $t_k$.

**Algorithm 3.** Obtaining $P_{\text{att}}$ at current time $t_k$.

1. If $A_{\text{start}}(k-1)$ is empty, which implies that $P_{\text{att}}(k-1)$ is $T(k-1)$, then
1.1. If $P_R T$ is not blocked by knowledge obstacle walls, then $P_{\text{att}}$ is $T$; $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ are empty; stop the algorithm. Otherwise, go to step 4.
2. Else
2.1. If $P_R(k-2)P_R$ has an intersection with $RL(A_{\text{start}}(k-1), A_{\text{end}}(k-1))$, then go to step 3; otherwise, go to step 2.2.
2.2. If $P_R A_{\text{end}}(k-1)+A_{\text{length}}(k-1)$ has an intersection with the segments in $S_{v1}$, then go to step 4; otherwise, go to step 2.3.
2.3. If $P_R A_{\text{end}}(k-1)$ is not blocked by the segments in $S_{v2}$, then $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ are same as that at $t_k - 1$; $P_{\text{att}}$ is $A_{\text{end}}$; stop the algorithm. Otherwise, go to step 2.4.

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

2.4. If $P_R A_{\text{start}}(k-1)$ is not blocked by the segments in $S_{v2}$, then $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ are same as that at $t_{k-1}$; $P_{\text{att}}$ is $A_{\text{start}}$; stop the algorithm. Otherwise, go to step 4.
2.5 End if
3. If $P_R T$ is not blocked by knowledge obstacle walls, then $P_{\text{att}}$ is $T$; $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$ are empty; stop the algorithm. Otherwise, go to step 4.
4. Use Algorithm 2 to set *B-type* artificial obstacle segments and to obtain $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$.
5. If $A_{\text{end}}$ is empty, which implies that $T$ is unreachable, then $P_{\text{att}}$ is empty; stop the algorithm. Otherwise, go to step 6.
6. If $P_R A_{\text{end}}$ is not blocked by the segments in $S_{v2}$, then $P_{\text{att}}$ is $A_{\text{end}}$. Otherwise, $P_{\text{att}}$ is $A_{\text{start}}$.

From Algorithm 3, we can find out that $A_{\text{end}}$ or $A_{\text{start}}$ will be selected as the attractive point $P_{\text{att}}$ in general. Using attractive segments for attractive points has advantages as follows:

1. Each attractive segment will be used for several sampling times in general, as discussed in Section 3.2 and Algorithm 3. This feature can not only decrease the computation for $P_{\text{att}}$ greatly, but also prevent $P_{\text{att}}$ from being changed frequently.
2. $A_{\text{end}}$ will be selected as $P_{\text{att}}$ as much as possible. As such, $R$'s security is guaranteed to a certain extent and $R$'s traveling path will be smoother as $A_{\text{end}}$ is not on obstacles in general, as shown in Fig. 4.
3. Under the guidance of the attractive points, $R$'s traveling path can be optimized obviously. Furthermore, the global convergence is guaranteed to a certain extent, as shown in theorems below.

**Theorem 1.** *If (1) $T$ and obstacles are all static; (2) $R$ moves directly toward the attractive points obtained by Algorithm 3 at each step; (3) there is a time $t_{k1}$ such that no new points can be added to the knowledge obstacle walls and no artificial obstacle segment can be set for any time $t_k$ ($k > k_1$); (4) there is at least a feasible path for $R$ to reach $T$, then $R$ will reach $T$ in finite time.*

**Proof.** According to the given conditions, it can be concluded that if $P_R(k_1)T$ is not blocked by knowledge obstacle walls, then the theorem is true as $T$ will be selected as the attractive point and $R$ will move directly toward $T$ for all the times until reaching $T$.

Otherwise, a finite time $t_{k2}(k_2 > k_1)$ exists such that $P_R(k_2-2)P_R(k_2)$ has an intersection with $RL(A_{\text{start}}(k_1), A_{\text{end}}(k_1))$ and $P_R(k_2) = A_{\text{start}}(k_1)$ (almost).

If $P_R(k_2)T$ is not blocked by knowledge obstacle walls, then the theorem is true, as discussed previously. Otherwise, $A_{\text{start}}(k_2)$ is required to be obtained, which is different from $A_{\text{start}}(k_1)$ according to Algorithm 3. In the following, we will prove that $|P_R(k_2)T| + |FW(P_R(k_2) \sim A_{\text{start}}(k_2) \sim T)|$ will be less than $|P_R(k_1)T| + |FW(P_R(k_1) \sim A_{\text{start}}(k_1) \sim T)|$.

When $\Sigma_{\text{bw}}(k_2)$ is the same as $\Sigma_{\text{bw}}(k_1)$, assume that $FW(P_R(k_1) \sim A_{\text{start}}(k_1) \sim T)$ is $P_R(k_1)A_{\text{start}}(k_1)V_2, \ldots, V_nT$. Then $V_2$ is an open or close tangent point from $P_R(k_2)(= A_{\text{start}}(k_1))$ to $\Sigma_{\text{bw}}(k_2)$ and $|FW(P_R(k_2) \sim V_2 \sim T)| = |A_{\text{start}}(k_1)V_2, \ldots, V_nT|$. As

$$\left.\begin{aligned} &|P_R(k_2)T| + |A_{\text{start}}(k_1)V_2 \ldots V_nT| < \\ &|P_R(k_1)T| + |P_R(k_1)A_{\text{start}}(k_1)V_2 \ldots V_nT| \\ &|FW(P_R(k_2) \sim A_{\text{start}}(k_2) \sim T| \le |FW(P_R(k_2) \sim V_2 \sim T| \end{aligned}\right\}$$

$$\begin{aligned} \Rightarrow\ &|P_R(k_2)T| + |FW(P_R(k_2) \sim A_{\text{start}}(k_2) \sim T)| \\ &< |P_R(k_1)T| + |FW(P_R(k_1) \sim A_{\text{start}}(k_1) \sim T)|. \end{aligned}$$

If $\Sigma_{\text{bw}}(k_2)$ is different from $\Sigma_{\text{bw}}(k_1)$, then an open tangent point $Q$ from $P_R(k_2)(= A_{\text{start}}(k_1))$ to $\Sigma_{\text{bw}}(k_2)$ must exist such that $RL(A_{\text{start}}(k_1), Q)$ has an intersection $V$ with $P_R(k_1)T$, and $V$ is not on $A_{\text{start}}(k_1)Q$ (otherwise, an artificial obstacle segment set as $\Sigma_{\text{bw}}(k_2)$ will have an intersection with $P_R(k_1)T$), as shown in Fig. 5. As

$$\left.\begin{aligned} &|P_R(k_2)T| + |FW(P_R(k_2) \sim A_{\text{start}}(k_2) \sim T)| \\ &\le |P_R(k_2)T| + |FW(P_R(k_2) \sim Q \sim T)| \\ &|FW(P_R(k_2) \sim Q \sim T)| < |A_{\text{start}}(k_1)VT| \\ &|P_R(k_2)T| + |A_{\text{start}}(k_1)VT| < \\ &|P_R(k_1)T| + |P_R(k_1)A_{\text{start}}(k_1)T)| \end{aligned}\right\}$$

$$\begin{aligned} \Rightarrow\ &|P_R(k_2)T| + |FW(P_R(k_2) \sim A_{\text{start}}(k_2) \sim T)| \\ &< |P_R(k_1)T| + |FW(P_R(k_1) \sim A_{\text{start}}(k_1) \sim T)|. \end{aligned}$$

As $|P_R T| + |FW(P_R \sim A_{\text{start}} \sim T)|$ decreases strictly, then a finite time $t_{k3}$ exists such that if $k > k_3$, then $A_{\text{start}}(k)$ is empty and $P_R(k)T$ is not blocked by knowledge obstacle walls. Thus, the theorem is true. □

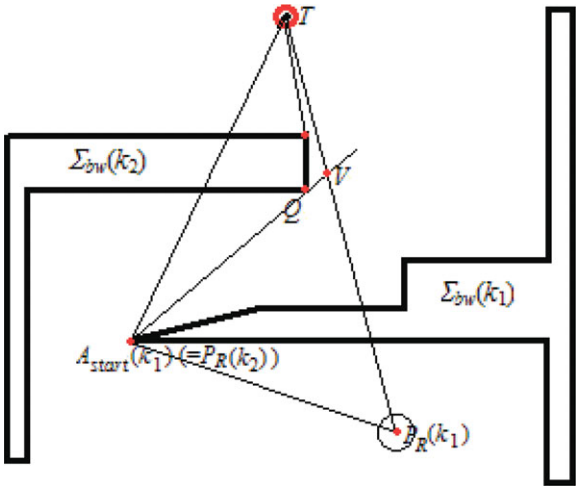



Fig. 5. (Colour online) $\Sigma_{\text{bw}}(k_2)$ is different from $\Sigma_{\text{bw}}(k_1)$.

**Theorem 2.** *If $T$ and obstacles are all static and $R$ moves almost directly toward the attractive points obtained*

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

*by Algorithm 3 at each step, then R will reach T or find out that no such path exists in finite time.*

**Proof.** For each period of time, if it is long enough, then new points must be added to the knowledge obstacle walls or new artificial obstacle segments must be set. Otherwise, if $T$ is reachable, then $R$ will reach $T$ within the times according to theorem 1.

In the worst case where all points on obstacles have been detected if possible and all knowledge obstacle walls have been connected together by artificial obstacle segments, no new point can be added to the knowledge obstacle walls and no new artificial obstacle segment can be set. Then, $T$ will be found out to be unreachable. Otherwise, $R$ will reach $T$ in finite time according to the aforementioned discussion. □

### *3.4. General steps for the proposed path planner*

The general steps of the proposed path planner are shown in Algorithm 4.

**Algorithm 4.** General steps for the proposed path planner.

1. Set the values of the parameters in the control system; initialize $R$, $T$, environment and knowledge obstacle walls; let $t_k = t_0$; $A_{\text{start}}(-1)$, $A_{\text{end}}(-1)$ and $A_{\text{length}}(-1)$ are empty.
2. Remove all artificial obstacle segments associated with the knowledge obstacle segments that are found in motion.
3. Set $N$-type artificial obstacle segments until no such segment can be set.
4. Using Algorithm 3 to obtain $P_{\text{att}}$, $A_{\text{start}}$, $A_{\text{end}}$ and $A_{\text{length}}$.
5. If $P_{\text{att}}$ is empty, then stop the algorithm and conclude that $T$ is unreachable. Otherwise, go to step 6.
6. $R$ moves a step to the next time $t_{k+1}$ under the guidance of $P_{\text{att}}$ and the control of the artificial moment motion controller introduced in Section 4.
7. If $R$ reaches $T$, then stop the algorithm; otherwise, update all knowledge obstacle walls and let $t_k = t_{k+1}$, and then return to step 2.

**Remark 2.** *Compared with the pure APF and "bug" algorithms, the proposed one is required to set/remove artificial obstacle segments and to compute attractive points, and then more computation may be needed. However, as the attractive points are obtained based on attractive segments and each attractive segment will be used for several sampling times in general, the computation burden of the proposed one can keep at a low level and can meet the requirements of real-time property.*

## 4. Motion Controller Based on Artificial Moments

A motion controller based on artificial moments for single-robot path planning is designed in this section, which is an extension of the artificial moment method for multi-robot formation control.[26,27] One basic idea of the controller is that at time $t_k$, $R$ is influenced by an attractive moment imposed by the attractive point $P_{\text{att}}(k)$ and some repulsive moments imposed by obstacles. Under the influence of the artificial moments, $R$ changes its PMDline and position along the gradient of the total moments so that the moments can increase quickly. As this idea is similar to that based on APF, the controller has the same advantages as that based on APF. However, there are two important differences between them.

One is that an additional motion vector, which is not influenced by the artificial moments, always exists in the proposed controller but not that based on APF. Let $(S_x(k+1), S_y(k+1))^T$ represent the expected additional motion vector in $(t_k, t_{k+1}]$. Then, if $P_{\text{att}}(k)$ is $T(k)$ and $|P_R(k)T(k)| \le D_S$,

$$\begin{cases} S_x(k+1) = S_T(k)\cos\beta_T(k) \\ S_y(k+1) = S_T(k)\sin\beta_T(k) \end{cases}. \tag{2}$$

Otherwise,

$$\begin{cases} S_x(k+1) = S_R\cos[\beta_R(k+1)] \\ S_y(k+1) = S_R\sin[\beta_R(k+1)] \end{cases}, \tag{3}$$

where $S_T(k)$, $\beta_T(k)$ are the step length and PMDline's direction angle of $T$ in $(t_{k-1}, t_k]$, respectively; $\beta_R(k+1)$ is the direction angle of $R$'s PMDline in $(t_k, t_{k+1}]$.

The motion vector $(S_x(k+1), S_y(k+1))^T$ plays an important role in the controller, as if $T$ is in motion, the vector can serve as a feed-forward compensation which makes it possible for $R$ to follow $T$ exactly. For obstacle avoidance, the vector makes $R$ have high speed even if the total moment's gradient is zero, and then it is impossible for $R$ to be trapped by a "local minimum" point.

The second difference is that in the APF, artificial potential functions are designed always in terms of the positions, velocities, and accelerations of $R$, $T$ and obstacles, or relative ones between them;[15–17] artificial moments, however, are designed always in terms of the angles from $R$'s PMDline to the attractive points and obstacles, and in most cases, artificial moments influence only $R$'s PMDline but not $R$'s position and velocity. As a result, many problems encountered in the APF, such as no passage between closely spaced obstacles and goals non-reachable with obstacles nearby,[16] are solved by the proposed motion controller.

The details of the proposed motion controller are described as follows:

Assume that function amt$(x)$ is as (4), where $\delta_\theta$ is a positive constant far less than $\pi/2$.

$$\text{amt}(x) = \begin{cases} \cos(\delta_\theta) + (\delta_\theta^2 - x^2)/2 & |x| \le \delta_\theta \\ \cos(x) & \delta_\theta < |x| \le \pi/2 \\ \dfrac{\pi}{2} - |x| + \cos\left(x - \dfrac{\pi}{2}\text{sign}(x)\right) & |x| > \frac{\pi}{2} \end{cases}. \tag{4}$$

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

Then, the derivative damt($x$) of amt($x$) is as (5).

$$\text{damt}(x) = \begin{cases} -x & |x| \le \delta_\theta \\ -\sin x & \delta_\theta < |x| \le \pi/2 \\ -\text{sign}(x) - \sin\left(x - \frac{\pi}{2}\text{sign}(x)\right) & |x| > \frac{\pi}{2} \end{cases}. \tag{5}$$

Let $AM(k)$ denote the attractive moment generated by the attractive point $P_{\text{att}}(k)$. Then, when $P_{\text{att}}(k)$ is not $T$ or $|P_R(k)T(k)| > D_S$, $AM(k)$ is only required to make $R$'s PMDline face toward $P_{\text{att}}(k)$ for the reason that if $R$'s PMDline is face toward $P_{\text{att}}(k)$, then the motion vector $(S_x(k+1), S_y(k+1))^T$ will make $R$ close to $P_{\text{att}}(k)$. As such, $AM(k)$ is designed as (6) in this case.

$$AM(k) = \frac{\lambda_a}{2}[\text{agl}(\beta_R(k) - \beta(P_R(k), \quad P_{\text{att}}(k)))]^2, \tag{6}$$

where $\lambda_a$ is a constant and $0.5 < \lambda_a < 1$.

When $P_{\text{att}}(k)$ is $T$ and $|P_R(k)T(k)| \le D_S$, $AM(k)$ is required to make both the PMDline direction and position of $R$ be as same as that of $T$. Thus, $AM(k)$ is designed as (7) in this case.

$$\begin{aligned} AM(k) = {} & \text{amt}[\text{agl}(\beta_R(k) - \beta_T(k))] \\ & + \left(\frac{D_S}{\pi}\right)^2 \left[\text{amt}\left(\frac{\pi}{D_s}(x_R(k) - x_T(k))\right)\right. \\ & \left. + \text{amt}\left(\frac{\pi}{D_s}(y_R(k) - y_T(k))\right)\right]. \end{aligned} \tag{7}$$

Let $(\Delta_1\beta_R(k), \Delta_1 x_R(k), \Delta_1 y_R(k))^T$ denote the expected vector to increase $AM(k)$. Then, when $P_{\text{att}}(k)$ is not $T$ or $|P_R(k)T(k)| > D_S$, $(\Delta_1\beta_R(k), \Delta_1 x_R(k), \Delta_1 y_R(k))^T$ is as (8), which is the gradient of $AM(k)$ computed by (6).

$$\begin{aligned} & (\Delta_1\beta_R(k), \Delta_1 x_R(k), \Delta_1 y_R(k))^T \\ & \quad = (-\lambda_a[\text{agl}(\beta_R(k) - \beta(P_R(k), \ P_{\text{att}}(k)))], \ 0, \ 0)^T. \end{aligned} \tag{8}$$

When $P_{\text{att}}(k)$ is $T$ and $|P_R(k)T(k)| \le D_S$, $(\Delta_1\beta_R(k), \Delta_1 x_R(k), \Delta_1 y_R(k))^T$ is as (9), which is the gradient of $AM(k)$ computed by (7).

$$\begin{pmatrix} \Delta_1\beta_R(k) \\ \Delta_1 x_R(k) \\ \Delta_1 y_R(k) \end{pmatrix} = \begin{pmatrix} \text{damt}[\text{agl}(\beta_R(k) - \beta_T(k))] \\ \frac{D_S}{\pi}\text{damt}[\frac{\pi}{D_S}(x_R(k) - x_T(k))] \\ \frac{D_S}{\pi}\text{damt}[\frac{\pi}{D_S}(y_R(k) - y_T(k))] \end{pmatrix}. \tag{9}$$

Each dangerous wall representative point will, and only such points can, impose a repulsive moment on $R$ by which $R$ is forced away from obstacles. Suppose that $G$ is a dangerous wall representative point at $t_k$, and $PM(k)$ is the repulsive moment imposed by $G$. Then, $PM(k)$ is only required to make $R$'s PMDline opposite to $G$ for the reason that the motion vector $(S_x(k+1), S_y(k+1))^T$ will make $R$ away from $G$ when $R$'s PMDline is opposite to $G$. As such, $PM(k)$ is designed as (10).

$$\begin{aligned} PM(k) = -\frac{\lambda_p}{2} & \left[\text{agl}(\text{agl}(\beta_R(k) + \Delta_1\beta_R(k)) \right. \\ & \left. - \beta(G, \ P_R(k)))\right]^2, \end{aligned} \tag{10}$$

where

$$\lambda_p = \frac{D_S - |P_R(k)G|}{D_S}.$$

**Remark 3.** *The reason for using* $\text{agl}(\text{agl}(\beta_R(k) + \Delta_1\beta_R(k)) - \beta(G, P_R(k))$ *but not* $\text{agl}(\beta_R(k) - \beta(G, P_R(k))$ *to compute PM(k) is to avoid the effects of the repulsive and attractive moments to be cancelled out by each other, as when they are cancelled out by each other, not only R's safety cannot be guaranteed, but also R may move away from* $P_{\text{att}}(k)$.

**Remark 4.** *Obviously,* $0 < \lambda_p < 1$, *and* $\lambda_p$ *will increase with the decrease of* $|P_R G|$. *The reason is that the less the*$|P_R G|$ *is, the larger the repulsive moment is required to be, so that R's PMDline can be opposite to G quickly.*

Let $\Delta_2\beta_R(k)$ denote the expected vector to increase $PM(k)$, then it is as (11), which is the gradient of $PM(k)$.

$$\Delta_2\beta_R(k) = -\lambda_p \text{agl}(\text{agl}(\beta_R(k) + \Delta_1\beta_R(k)) - \beta(G, P_R(k))). \tag{11}$$

The sum of all repulsive moments' gradients is denoted by $\Sigma\Delta_2\beta_R(k)$, where when no dangerous wall representative point exists, $\Sigma\Delta_2\beta_R(k) = 0$.

The controller based on artificial moments for single-robot path planning is then designed as (12)–(15):

$$\beta_R(k+1) = \text{agl}[\beta_R(k) + \Delta_1\beta_R(k) + \sum \Delta_2\beta_R(k)]. \tag{12}$$

Let $S_L$ denote $R$'s expected step length, and then

$$S_L = \sqrt{(S_x(k+1) + \Delta_1 x_R(k))^2 + (S_y(k+1) + \Delta_1 y_R(k))^2}. \tag{13}$$

If $S_L$ is not larger than $S_R$, then

$$\begin{cases} x_R(k+1) = x_R(k) + S_x(k+1) + \Delta_1 x_R(k) \\ y_R(k+1) = y_R(k) + S_y(k+1) + \Delta_1 y_R(k) \end{cases}. \tag{14}$$

Otherwise, $(x_R(k+1), y_R(k+1))^T$ is computed by (15) so that $R$'s step length will not be larger than $S_R$:

$$\begin{cases} x_R(k+1) = x_R(k) + \frac{S_R}{S_L}(S_x(k+1) + \Delta_1 x_R(k)) \\ y_R(k+1) = y_R(k) + \frac{S_R}{S_L}(S_y(k+1) + \Delta_1 y_R(k)) \end{cases}. \tag{15}$$

From (2), (10)–(15), we have that when $P_{\text{att}}(k)$ is not $T$ or $|P_R(k)T(k)| > D_S$, $R$'s moving speed is always $S_R$, the longest step length of $R$. Therefore, it is impossible for $R$

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

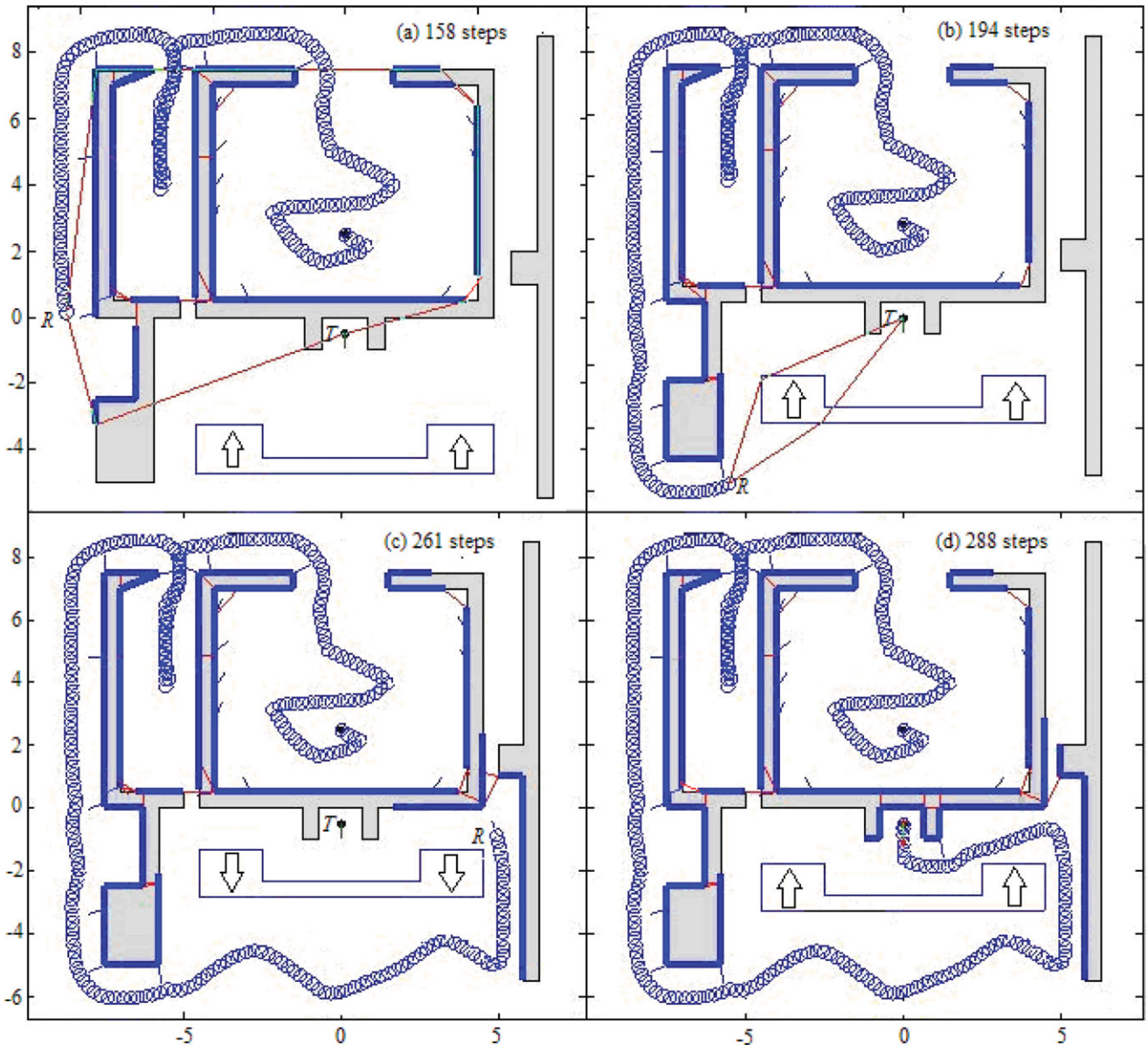



Fig. 6. (Colour online) Path planning by the proposed path planner in the first environment.

to stay at a "local minimum" point or a small region in the motion controller, that is to say, $R$ is difficult to be trapped even in complicated environments.

Compared with the motion controller based on artificial moments for formation control,[26,27] the implementation of the controller in this paper is simpler for the simpler artificial moments and the less number of parameters.

## 5. Simulations and Analysis

Extensive simulations have been carried out by using the proposed path planner, and two of them are given in Figs. 6, 9 and 10, respectively, where short blue segments on obstacles are attractive segments obtained at different times. The parameters are $D_R = 0.2$, $D_V = 3.5$, $D_S = 1.2$, $S_R = 0.24$, $\lambda_a = 0.8$ and $\delta_\theta = \pi/90$. The speeds of $T$ and dynamic obstacles are 0.15 if they are in motion.

As a comparison, in almost the same situations and with almost the same parameters, the results using the improved APF presented in ref. [19] are also given in Figs. 7, 8 and 10, respectively, where the wall following method (WFM) is used to solve the "local minimum" problem in the APF.[19] $A_1, A_2, \ldots$ are the "local minimum" points where $R$ switches from APF mode to WFM mode; $B_1, B_2, \ldots$ are the points where $R$ switches from WFM mode to APF mode.

The first simulation environment is shown in Figs. 6 and 7, where an obstacle keeps moving up and down in a given region, and $T$ is static with obstacles nearby.

From Figs. 6 and 7, we can see that the proposed path planner makes $R$ bypass safely all obstacles and reach $T$ exactly in 288 steps. Meanwhile, no point similar to the "local minimum" point is encountered. The improved APF, however, cannot make $R$ reach $T$ as at the "local minimum" point $A_3$, as shown in Fig. 7(b), $R$ cannot switch from APF mode into WFM mode successfully for the reason that $T$ is found to be not blocked by obstacles, and then no obstacle will be selected for $R$ to follow. Moreover, in the avoidance of the dynamical obstacle, the generated path in the proposed

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

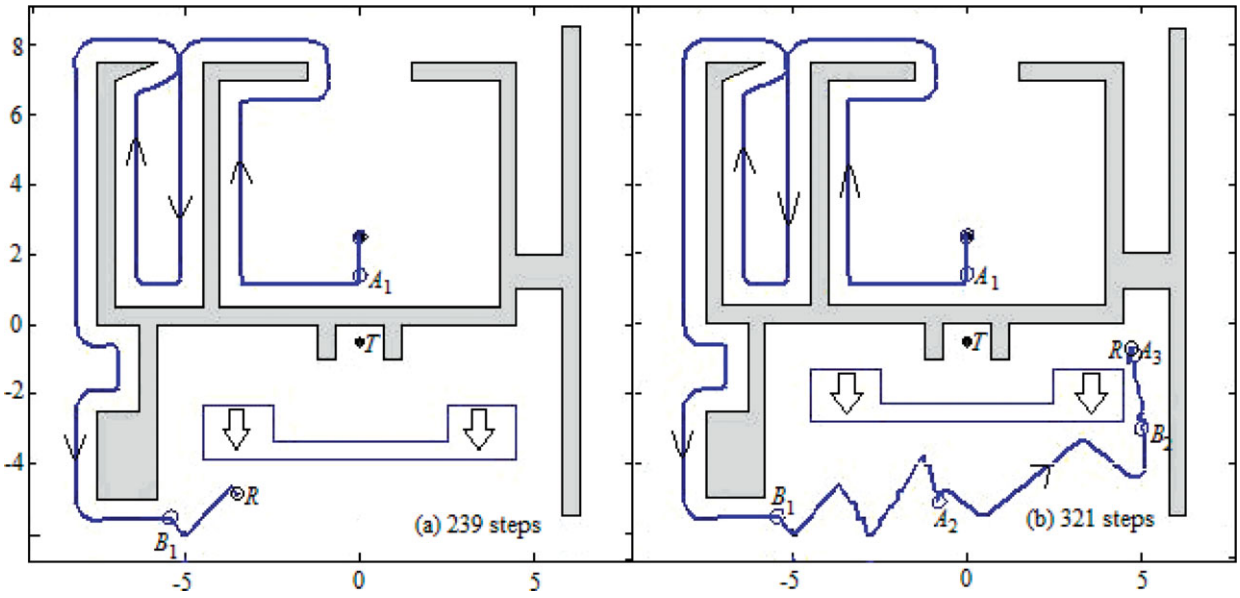



Fig. 7. (Colour online) Path planning by the improved APF in the first environment.

one is smoother and shorter than that in the improved APF, as shown in Figs. 6(c), (d) and 7(b).

The second simulation environment is shown in Figs. 8 and 9, where an obstacle moves up firstly to a given position, and then goes back and keeps staying at its original position. $T$ is close to and is surrounded by an obstacle. Different from the $T$ in the first environment, $T$ in the second one keeps moving up and down in a given region.

From Figs. 9 and 10(a), we can see that the proposed path planner makes $R$ bypass all obstacles safely and pass through narrow passage successfully and smoothly as the WFM. At steps 404, $R$ has reached $T$ exactly. Subsequently, the distance between $P_R$ and $T$ remains zero, which implies that the proposed algorithm can make $R$ follow a dynamical target exactly.

From Figs. 8 and 10(b), we can see that the improved APF also makes $R$ bypass all obstacles and pass through narrow passage. However, the generated path is much longer than that in the proposed one. Furthermore, $R$ cannot reach $T$ exactly, as shown in Fig. 10(b), which implies that the improved APF cannot make $R$ follow a dynamical target exactly.

For the computation of the proposed path planner, it is almost the same as that of the improved APF, except for some special times where the computation of the proposed one is somewhat more than that of the improved APF.

These simulations verify, compared with the APF and the WFM, that the proposed one is more effective in aspects such as path optimization, dynamical obstacle avoidance, following dynamical targets and solving the problems similar

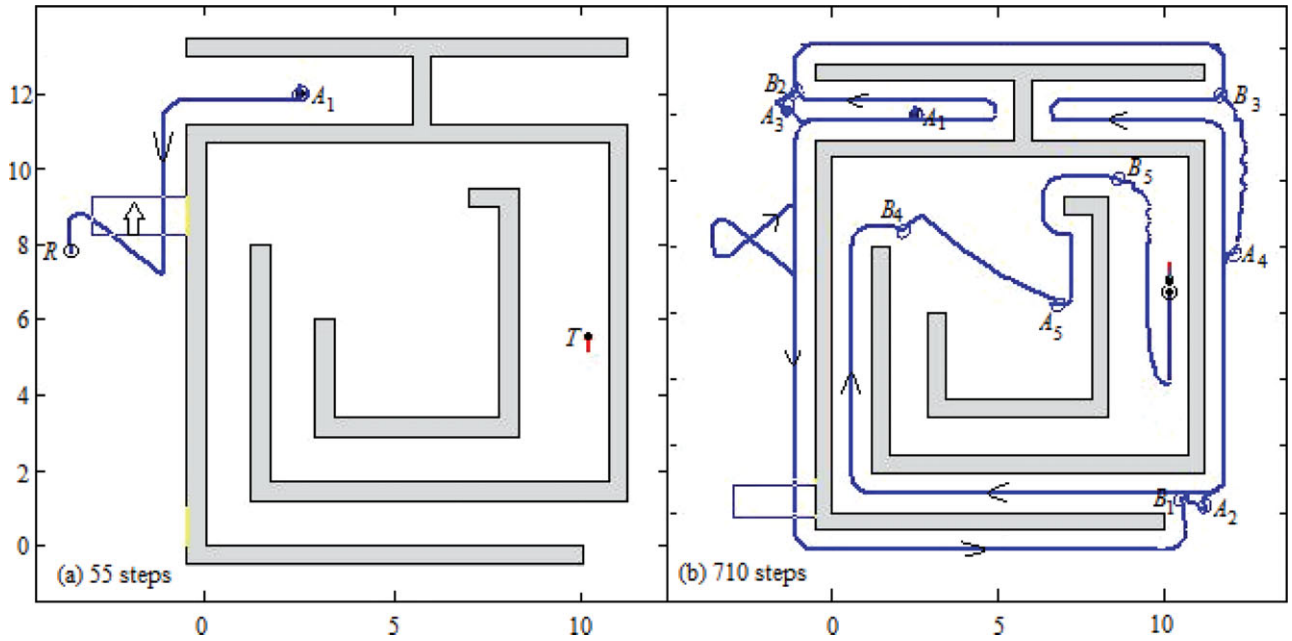



Fig. 8. (Colour online) Path planning by the improved APF in the second environment.

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

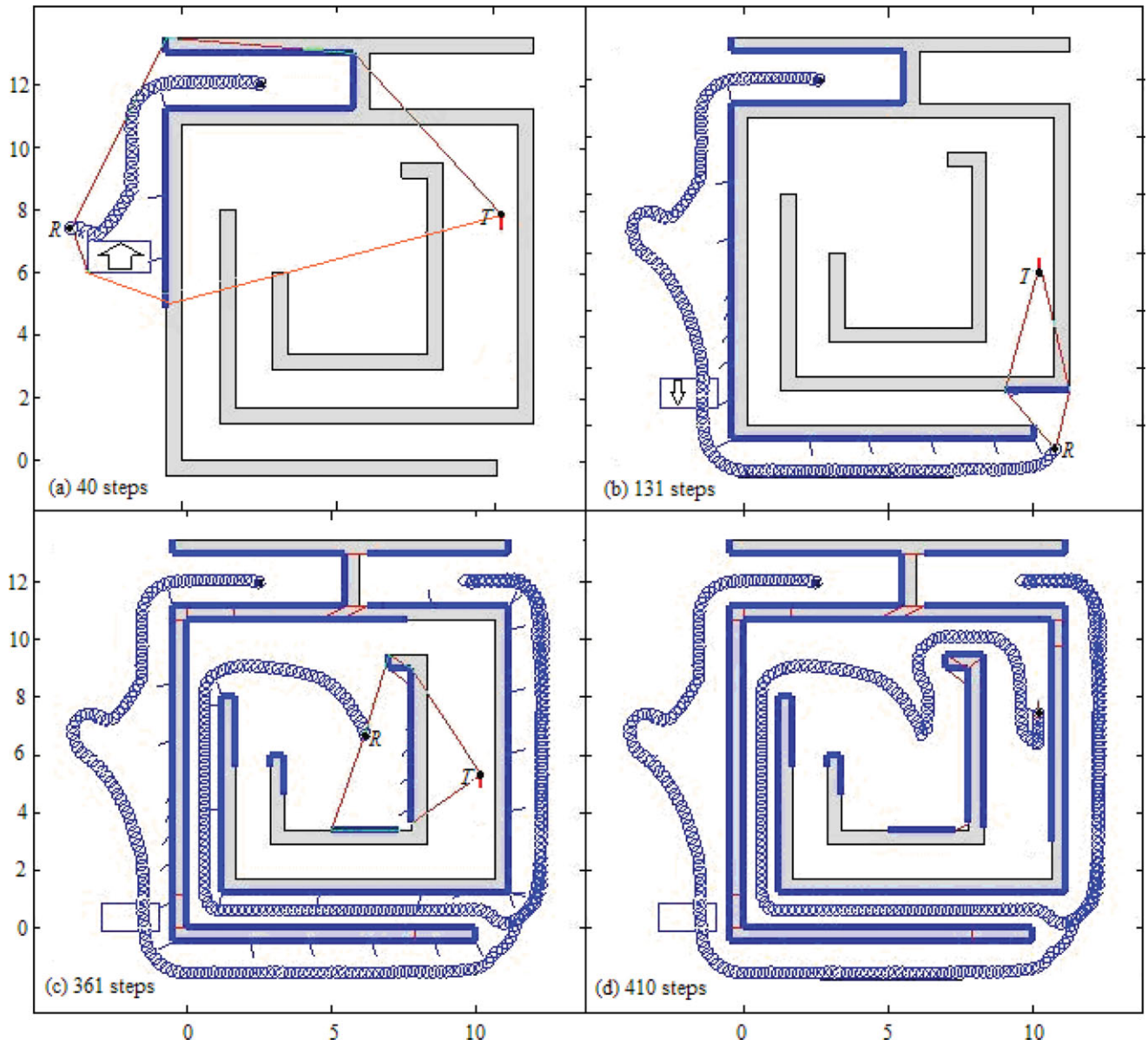



Fig. 9. (Colour online) Path planning by the proposed path planner in the second environment.

to the "local minimum" problem. Then, the proposed one is valuable.

## 6. Conclusions

The following aspects of the proposed path planner deserve attention:

1. *Attractive segments*. Using attractive segments for attractive points has advantages as follows. Firstly, each attractive segment is obtained by using the full but not local environmental knowledge. As such, better solutions are often to be obtained. Secondly, each attractive segment will be used for several sampling times until one of the conditions for changing attractive segments is satisfied. As such, $R$'s motion direction can be avoided to be changed frequently and the computation burden for attractive segments can keep at a low level. Thirdly, $A_{\text{end}}$ will be selected as $P_{\text{att}}$ as much as possible. As such, $R$'s security is guaranteed to a certain extent and $R$'s traveling path may be smoother as $A_{\text{end}}$ is not on obstacles in general.
2. *The motion controller based on artificial moments*. The controller has the same advantages as the APF method because they are similar in some aspects. These shared advantages can make $R$ move under the guidance of the attractive points even if the points are on obstacles, and can make $R$ avoid dynamical obstacles while no additional measures are taken. On the other hand, almost all problems encountered in the APF method, such as the "local minimum" problem, no passage between closely spaced obstacles and goals non-reachable with obstacles nearby, are solved by the controller. Furthermore, the controller can make $R$ have the desired posture as much as possible when $R$ reaches $T$. Therefore, the controller is valuable.

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press

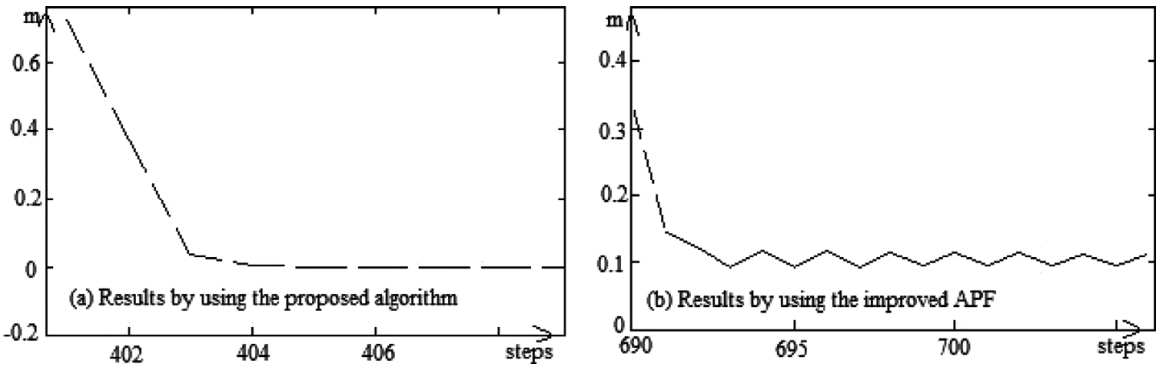



Fig. 10. Distance curves between $P_R$ and $T$ when $P_R$ is close to $T$ in the second environment.

An interesting topic for future work is to extend the proposed path planner to the multi-robot path planning in complicated dynamical environments.

**Acknowledgements**

This research was supported by the NSF of China under grant 60874017, the State Key Laboratory of Robotics and System (HIT) under grant SKLRS-2011-MS-03, and the Special Research Foundation of the Liaoning University of Science and Technology under grant 2012RC01.

**References**

1. D. Hsu, J. C. Latombe and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comput. Geom. Appl.* **9**(4& 5), 495–512 (1999).
2. A. C. Nearchou, "Path planning of a mobile robot using genetic heuristics," *Robotica* **16**, 575–588 (1998).
3. C. Undeger and F. Polat, "Real-time edge follow: A real-time path search approach," *IEEE Trans. Syst. Man Cybern. Part* C*: Appl. Rev.* **37**(5), 860–872 (2007).
4. S. S. Ge, X. Lai and A. A. Mamum, "Boundary following and globally convergent path using instant goals," *IEEE Trans. Syst. Man Cybern. Part* B*: Cybern.* **35**(2), 240–254 (2005).
5. M. A. Salichs and L. Moreno, "Navigation of mobile robots: Open questions," *Robotica* **18**, 227–234 (2000).
6. H. P. Huang and S. Y. Chung, "Dynamic Visibility Graph for Path Planning," **In**: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan (2004) pp. 2813–2818.
7. O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams", *IEEE Trans. Robot. Autom.* **5**(2), 143–150 (1989).
8. A. Sud, E. Andersen, S. Curtis, M. C. Lin and D. Manocha, "Real-time path planning in dynamic virtual environments using multiagent navigation graphs," *IEEE Trans. Vis. Comput. Graph.* **14**(3), 526–538 (2008).
9. Y. H. Liu and S. Arimoto, "Proposal of Tangent Graph and Extended Tangent Graph for Path Planning of Mobile Robots," **In**: *Proceedings of IEEE International Conference on Robotics and Automation,* Sacramento, CA (1991) pp. 312–317.
10. I. Kamon, E. Rimon and E. Rivlin, "Tangentbug: A range-sensor-based navigation algorithm," *Int. J. Robot. Res.* **17**(9), 934–953 (1998).
11. L. E. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).
12. P. Cheng and S. M. Lavalle, " Resolution Complete Rapidly-Exploring Random Trees," **In**: *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, America (2002) pp. 267–272.
13. J. Yakey, S. M. Lavalle and L. E. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Trans. Robot. Autom.* **17**(6), 951–958 (2001).
14. E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992).
15. A. Oustaloup, B. Orsoni, P. Melchior and H. Linarès, "Path planning by fractional differentiation," *Robotica* **21**(1), 59–69 (2003).
16. S. S. Ge and Y. J. Cui. "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.* **16**(5), 615–620 (2000).
17. L. Yin, Y. X. Yin and C. J. Lin, "A new potential field method for mobile robot path planning in the dynamic environments", *Asian J. Control* **11**(2), 214–225 (2009).
18. D. H. Kim, "Escaping route method for a trap situation in local path planning," *Int. J. Control Autom. Syst.* **7**(3), 495–500 (2009).
19. Y. Zhu, T. Zhang and J. Y. Song, "An Improved Wall Following Method for Escaping From Local Minimum in Artificial Potential Field Based Path Planning," **In**: *Proceedings of Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China (2009) pp. 6017–6022.
20. J. Ng and T. Bräunl, "Performance comparison of bug navigation algorithms," *J. Intell. Robot. Syst.* **50**, 73–84 (2007).
21. E. Besada-Portas, L. Torre, J. M. Cruz and B. Andrés-Toro, "Evolutionary trajectory planner for multiple UAVs in realistic scenarios," *IEEE Trans. Robot.* **26**(4), 619–634 (2010).
22. C. G. Zhang and Y. G. Xi, "Robot path planning in globally unknown environments based on rolling windows," *Sci. China-Ser.* E **44**(2), 131–139 (2001).
23. C. G. Zhang and Y. G. Xi, "Rolling path planning and safety analysis of mobile robot in dynamic uncertain environment," *Control Theory Appl.* (in Chinese) **20**(1), 37–44 (2003).
24. C. G. Zhang and Y. G. Xi, "A real-time path planning method for mobile robot avoiding oscillation and dead circulation," *Acta Autom. Sin.* **29**(2), 197–205 (2003).
25. C. G. Zhang and Y. G. Xi, "Sub-optimality analysis of mobile robot rolling path planning," *Sci. China-Series* F *Inf. Sci.* **46**(2), 116–125 (2003).
26. W. B. Xu and X. B. Chen, "Artificial moment method for swarm-robot formation control," *Sci. China-Ser.* F*: Inf. Sci.* **51**(10), 1521–1531 (2008).
27. W. B. Xu and X. B. Chen, "A dynamical formation control approach based on artificial moments," *Control Theory Appl.* (in Chinese) **26**(11), 1232–1238 (2009).

https://doi.org/10.1017/S0263574713000477 Published online by Cambridge University Press