The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Shape-shifting Robot Path Planning Method Based on Reconfiguration Performance

Tonglin Liu, Chengdong Wu and Bin Li

***Abstract*—A shape-shifting robot "AMOEBA-I" has diverse configurations, and the accessibility of the robot can be reinforced in the narrow space by changing the configurations. In this paper, a path planning method is presented corresponding to the unique reconfiguration ability of this robot. This method can automatically adjust the relation between the rapid movement and the secure mobile position of the robot based on the distribution of obstacles that are around the robot, and this method can also automatically select an appropriate configuration to adapt the robot to the environmental variation according to the information of the current conditions. The problem of deadlock is avoided by using the *Boundary Following*. Further, a *Reconfiguration Memory* is provided to optimize the trace of the *Boundary Following*, and it can help the robot to search a new path. Simulation results validated the advantage of the proposed method which can get the best out of the unique accessibility of the shape-shifting robot and reduced effectively the length of the robot traveling path.**

I. Introduction

DISASTERS happen frequently for the reason of earthquake, terrorist activities, and so on. If survivors can be searched and received duly treatment after the disaster occurred within 24 hours, the survival rate will be higher. Therefore, the rescue activities should be started quickly. Robots can effectively help people deal with affairs in the dangerous environment, and they can reduce kinds of risks that the associated personnel might suffer. A shape-shifting robot with automatic reconfiguration ability can complete the search and rescue task by changing the configurations to adapt to the dangerous, unknown and non-structural environment. Therefore, the foreground of shape-shifting robot has attracted a lot of researches.

One of the interested research topics in this area is path planning, which determines whether a robot can achieve designated tasks in time in the complex environment. All path planning approaches aim to find a barrier-free path from an actual position S of a controlled robot to a desired goal position G [1].

Two main approaches for solving the path planning problem are global and local methods. The global methods need complete information of the environment and they are based on a global representation. Examples of this approach are the grid of visits [2], the visibility graph [3], the behavior-based models [4], the ant colony algorithm [5] and the genetic algorithm [6, 7], etc.

On the other hand, the local path planners use only the local information in a purely reactive manner. Different methods are used to choose the next action according to the local environment: the bug algorithm [8], the fuzzy logic [9], the artificial potential field method [10] and the behavior-based models [11], etc.

For the need of disaster relief, AMOEBA-I [12, 13] is designed, which is a three-module shape-shifting robot. According to the characteristic of automatic reconfiguration, the shape-shifting robot can be considered as one that holds the reconfiguration ability to strengthen the accessibility. In this paper, we present a new path planning method that uses the reconfiguration ability of robot. The unique accessibility of shape-shifting robot is fully displayed. The deadlock condition is resolved by using the *Boundary Following*. Experiment results show that the robot can change its own configuration to adapt to current environment, and the reconfiguration situation is stored through the *Reconfiguration Memory*. Thus, the proposed method can effectively reduce the *Boundary Following* and shorten the length of the path.

II. The Shape-shifting Robot System

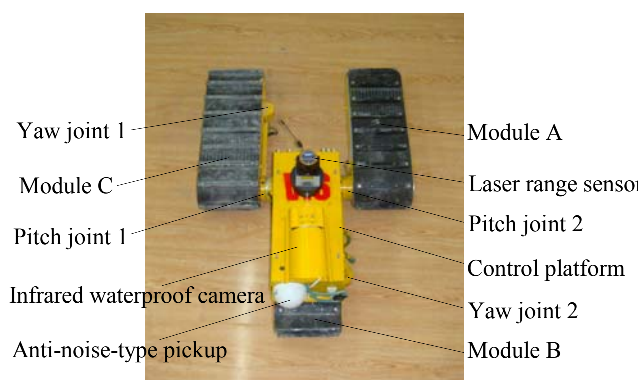


Fig. 1. A three-module shape-shifting robot, AMOEBA-I

Manuscript received March 10, 2010. This work is supported by the National Natural Science Foundation of China (Grant No. 60705029), the National High-Technology 863 Program (Grant No. 2007AA041502-5), Doctoral Startup Foundation of Liaoning Province and State Key Laboratory of Robotics foundation (Grant No. RLO200813).

Tonglin Liu is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China and Graduate School of the Chinese Academy of Sciences, Beijing 100039, China. (e-mail: ltl2006@126.com).

Chengdong Wu is with the School of Information Science & Engineering, Northeast University, Shenyang 110004, China. He is also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. (e-mail: wuchengdong@ise.neu.edu.cn).

Bin Li is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. (e-mail: libin@sia.cn).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

The shape-shifting robot is a link-type structure with three modules. This kind of link-type robot can automatically change configuration to pose various kinds of symmetry configurations and trim configurations, especially being in line or in row. The platform AMOEBA-I is showed in Fig. 1. An infrared waterproof camera, an anti-noise-type pickup, a laser range sensor, an electron compass, a GPS and an inclinometer are located on the control platform.

AMOEBA-I has nine kinds of configurations, and each kind of configuration has its own advantages. Fig. 2 shows two representative configurations.

Configuration "T" Configuration "d"

Fig. 2. Two representative configurations of the shape-shifting robot

III. Collision-free Path Planning

In the path planning, how to effectively use the appreciable surrounding environment information is crucial to find a path satisfying certain requirements. The proposed path planning method is a kind of optimization algorithm. It can find a better path according to the optimal function, and it can automatically adjust oneself corresponding to the number of obstacles in surrounding environment. Thus, the method can enable the robot to quickly and safely reach target point by using less power consumption.

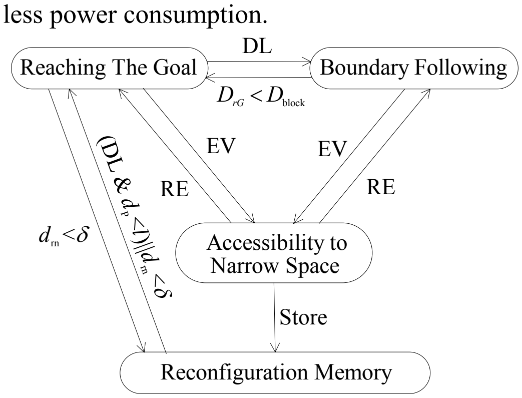


Fig.3. State transition diagram of the collision-free path planning

As shown in Fig. 3 the proposed method is composed of four states: *Reaching The Goal*, *Boundary Following*, *Accessibility to Narrow Space*, *Reconfiguration Memory*. In the unknown environment algorithm is first carried out from *Goal Touching*. When the deadlock occurs *Boundary Following* is implemented. Here DL is the abbreviation for deadlock. After *Reaching The Goal* or *Boundary Following* is implemented, *Accessibility to Narrow Space* will be executed if the environment varies. Here EV means environmental variation, RE means reconfiguration end. When the *Accessibility to Narrow Space* is implemented, the corresponding configuration is stored in *Reconfiguration Memory*. The *Reconfiguration Memory* will affect the change of goal position. Here d_P is the distance between the local deadlock position and P_{mi}, d_m is the distance between the robot and new target. l and δ are two coefficients.

A. Reaching The Goal

The candidate position of the robot is formulated as:

$$x_i(k) = q_{i-1} + \Delta t\dot{x} \tag{1}$$

Where $x_i(k)$ is kth candidate position, k=1, 2, …, c_3; and $\Delta t\dot{x}$ can be written in polar coordinates form as follows:

$$\Delta t\dot{x} = \begin{bmatrix} \lambda \\ \theta_k \end{bmatrix} \tag{2}$$

Where

$$\lambda = c_1 \min\left|q_{i-1} - p_j\right| + c_2 \min\left|q_{i-1} - p_j\right|(2rand - 1)$$

$$\theta_k = \theta_{k-1} + \frac{2\pi}{c_3} + \frac{\pi}{c_4}(2rand - 1)$$

λ is adaptation factor, as shown in Fig. 4. The closest sensory measurement is used from the obstacle surface. In the detection area of sensors, obstacles are scanned by sensor every certain angle. Let the detected points on the outline of obstacles be p_j(j=1, 2, …, n). min$|q_{i-1}-p_j|$ is the minimum distance between the robot center point and detected obstacle point. The proportional coefficient c_1 decides the basic length of adaptation factor λ, its value affects the calculation cost of the whole path planning algorithm. For considering the sensor errors in a real environment, a disturbance coefficient c_2 is introduced for adaptation factor λ. θ_k decides position of $x_i(k)$, and it starts with forward direction of the robot in position q_{i-1} in order to increase the probability of existence of the candidate position in the forward direction of robot, as shown in Fig. 4. c_3 determines the number of candidate position, c_4 is a disturbance coefficient of position. *rand* is random number within (0, 1).

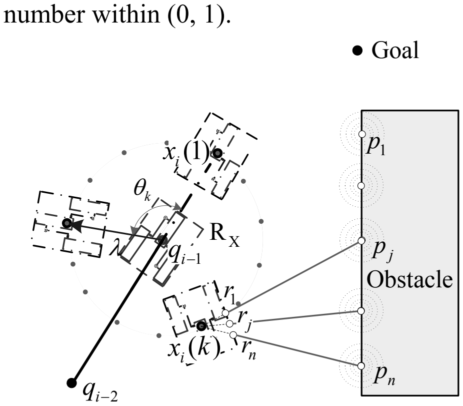


Fig. 4. Robot position determination

In order to ensure the robot can accurately reach the target position, adaptation factor λ is subject to the following constraints:

$$\lambda = \min(\lambda, |x_{\text{goal}} - q_{i-1}|) \tag{3}$$

Where $|x_{\text{goal}}-q_{i-1}|$ is distance between robot's position q_{i-1} and target position x_{goal}.

After determining the c_3 candidate positions, the $x_i(k)p_j$ represents a line segment between the point $x_i(k)$ and point p_i, and r_i is the intersections of line segment $x_i(k)p_j$ and R_X. The rectangle R_X whose length and width is determined by robot's current configuration represents the robot's simplified shape, as shown in Fig. 4. $d_{Xk}(j)$ (j=1, 2, …, n) is distance between point r_j and point p_j, and has been arranged from ascending order.

$$d_{Xk}(j) = |r_j - p_j| \quad d_{Xk}(j) < d_{Xk}(j+1) \quad (4)$$

If $d_{Xk}(1) < d_s$, the corresponding candidate position $x_i(k)$ will be abolished the candidate qualification (d_s is a safe distance).

In order to determine the forecast position q_i of the robot in the remaining m candidate positions, here an optimization function is defined as:

$$f = \min(K\frac{1}{1+D_k^2} + d_{\text{goal}}(k)) \quad (k=1,2,\ldots,m) \quad (5)$$

Where K is distance factor, it affects the distance between robot and obstacles. $D_k = d_{Xk}(1)$. $d_{\text{goal}}(k)$ is the distance between candidate position $x_i(k)$ and target position x_{goal}.

$$d_{\text{goal}}(k) = |x_i(k) - x_{\text{goal}}| \quad (k=1,2,\ldots,m) \quad (6)$$

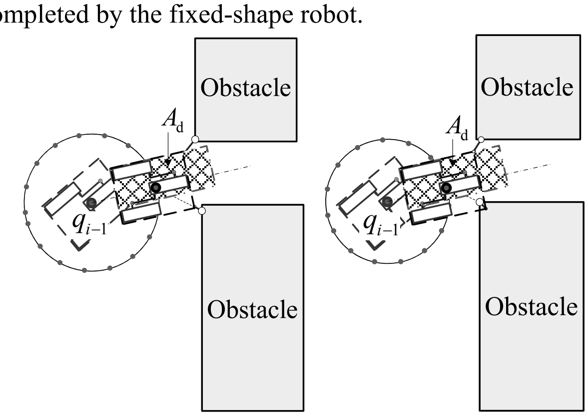
The robot's forecast position is formulated as:

$$q_i = f(x_i(k)) + (rand - \frac{1}{2}) \quad (7)$$

Where $f(x_i(k))$ shows that the optimal position is obtained by optimization function f in candidate positions. *rand* is random interference within [0, 1].

B. Accessibility to Narrow Space

So far, the method can assure that the fixed-shape robot can complete the path planning task according to the requirements. Here we consider the shape-shifting robot how to pass through narrow space by using the advantage of the reconfiguration. It will complete the task which can not be completed by the fixed-shape robot.



(a) (b)

Fig. 5. Situation of passing through the narrow space

Taking transformation between configuration "T" and configuration "d" for example, if the robot can pass through the narrow space by reconfiguration, two situations may happen in this candidate position shown in Fig. 5. One is the robot's shape near the obstacles and there are no obstacles in region A_d, as shown in Fig. 5(a). The other is the robot's shape and the shape of obstacles overlap, and there are no obstacles in region A_d, as shown in Fig. 5(b). The length and the width of the region A_d are defined as:

$$\begin{cases} l_A \geq l_d + d_s \\ b_A \geq b_d \end{cases} \quad (8)$$

Where l_d and b_d are the length and the width of configuration "d" respectively.

In both cases of Fig. 5 the robot can pass through the narrow space by reconfiguration, so the corresponding reconfiguration constraints as follows:

1. *if* $d_{XLk}(1) < d_s$ and $d_{XRk}(1) < d_s$ *then*
2. *if* $d_{YLk}(1) + d_{YRk}(1) \geq d_s$ and $\{p_{Li}, p_{Ri}\} \notin A_d$ *then*

$$D_k = \frac{1}{2}(d_{YLk}(1) + d_{YRk}(1))$$

3. *else*
4. The candidate qualification of the candidate position $x_i(k)$ is abolished.
5. *end*
6. *end*

In reconfiguration constraint, subscript X of $d_{XLk}(1)$ and $d_{XRk}(1)$ is the current configuration of the robot, subscript Y of $d_{YLk}(1)$ and $d_{YRk}(1)$ is the desired configuration of the robot. In Fig. 5 the shape-shifting robot changes configuration from configuration "T" to configuration "d", so subscript X shows configuration "T" and subscript Y shows configuration "d". According to (5), if the candidate position is identified as the forecast position q_i, the robot changes configuration in position q_{i-1}. When $d_{YLk}(1) \geq d_s$ and $d_{YRk}(1) \geq b_d$, the robot will change back to the original configuration.

C. Boundary Following

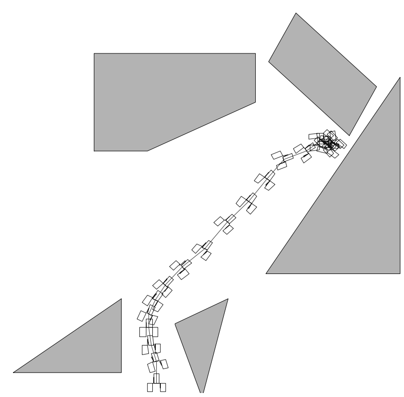


Fig. 6. The deadlock condition in path planning

An obvious phenomenon, which the robot falls into the deadlock, is that the direction of robot movement dramatically changes times without number, as shown in Fig. 6. The method for solving deadlock is as following:

Step 1: Judging the existence of the deadlock

The interval of robot's direction angle between in position $\boldsymbol{q}_{i-1}$ and in position $\boldsymbol{q}_i$ is supposed as $\triangle\alpha_t$. The robot falls into a deadlock if $\triangle\alpha_t \geq \varphi$, as shown in Fig. 7(a). The current value D_{block} of the distance function between deadlock position $\boldsymbol{q}_{i-1}$ and target position G is recorded. The value D_{block} decides the start and stop, that the robot follows the boundary of the obstacle.

Step 2: Method of escaping

After the deadlock situation is confirmed, the interval of the robot's direction angle is limited in $\triangle\alpha_t \in [-\varphi, \varphi]$, supposing φ=100^{o} in all experiments. A tracking distance d_{fo} between the robot and the boundary of obstacles is defined. In the position $\boldsymbol{q}_{i\text{-}1}$, if the minimum distance $d_{ro(i\text{-}1)}$ between the robot and the boundary of obstacles is kept in $[D_{fo}+\xi, D_{fo}-\xi]$, there is $d_{fo}=d_{ro(i\text{-}1)}$, otherwise there is $d_{fo}=D_{fo}$. D_{fo} is normal tracking distance between the robot and the boundary of obstacles, ξ is a small number which it allows the normal tracking distance varies in a small range. Therefore, the robot motion is reduced for searching appropriate tracking distance d_{fo}, and ξ must satisfy a constraint $D_{fo}-\xi \geqslant d_s$.

In the range of $\triangle\alpha_t\in[-\varphi, \varphi]$, a candidate position $x_i(k)$ with $\min|d_{ro(i)}(k)-d_{fo}|$ is found as the forecast position q_i. $d_{ro(i)}(k)$ is the minimum distance between the robot and the boundary of obstacles in the candidate position $x_i(k)$. If there are several equivalent minimum values, a candidate position which minimizes the $\triangle\alpha_t$ is selected as the forecast position q_i, which is shown in Fig. 7(b).

When the robot is circumventing the blocking obstacles, the distance D_{rG} between the current position of the robot and the target position is constantly calculated until a point, which is closer to the target than the deadlock position, can be found. At that time, the D_{rG} is less than D_{block} and the algorithm considers that the escape of the deadlock situation has been completed.

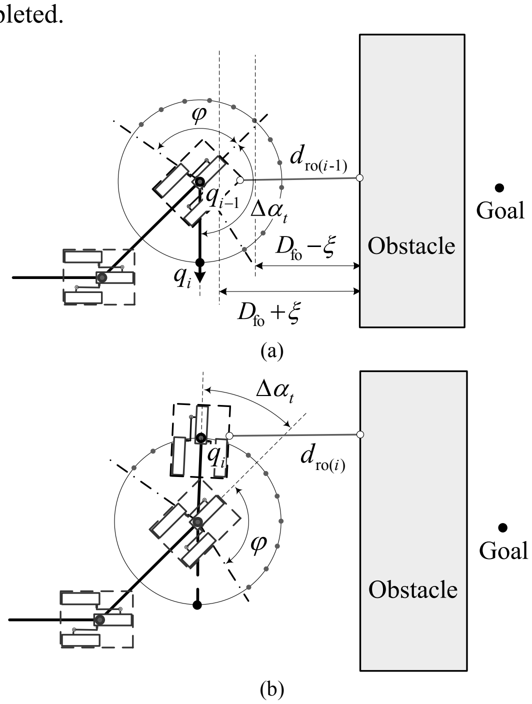


Fig. 7. Deadlock situation and boundary following

D. Reconfiguration Memory

The *Reconfiguration Memory* of the shape-shifting robot is that the situation, which the robot uses reconfiguration, is recorded according to the reconfiguration ability of the robot and the environmental characteristics when the robot encounters the narrow space. Taking the reconfiguration position as the circle center, if the deadlock situation occurs in a specified range, the robot can quickly return to this position to effectively escape from the deadlock region, and find a new path.

Taking two kinds of classical configurations for example, we mark the configuration of the robot with the binary code: the binary code of configuration "T" is 00, and the binary code of configuration "d" is 01. The storage table of the *Reconfiguration Memory* m_{ei} of the robot is shown in table I. In the *Reconfiguration Memory*, P_{mi} stores the corresponding reconfiguration position. C_{mi} stores the required configuration which the robot passes through the narrow space. η_{mi} stores the reconfiguration evaluation, which decides the robot whether to take reconfiguration.

TABLE I
RECONFIGURATION MEMORY OF ROBOT

	m_{e1}		…		m_{ei}	
P_{m1}	C_{m1}	η_{m1}	…	P_{mi}	C_{mi}	η_{mi}
(x_1, y_1)	01	1/0	…	(x_i, y_i)	01	1/0

When the shape-shifting robot encounters local deadlock situation, the escape rules of the robot is as follows:

1) When the distance between the local deadlock position and P_{mi} is less than l, P_{mi} is used as a new target position if there is *Reconfiguration Memory*. The target changes into the initial target G when the distance between the robot and new target is less than δ. If a local deadlock position is encountered without the *Reconfiguration Memory* before the robot reaches the new target. After the robot reaches the new target the reconfiguration evaluation η_{mi}=1, otherwise η_{mi}=0.
2) After the target changes into initial target G, the robot does not reconfiguration but chooses the other path if η_{mi}=1. Before the new *Reconfiguration Memory* is not stored, the path planning is done by using the *Boundary Following* method when the local deadlock situation is encountered.

IV. SIMULATIONS

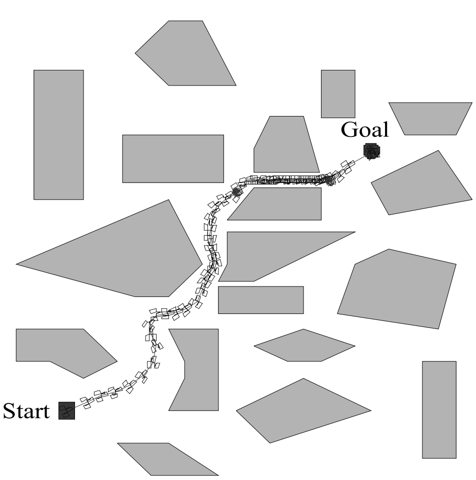
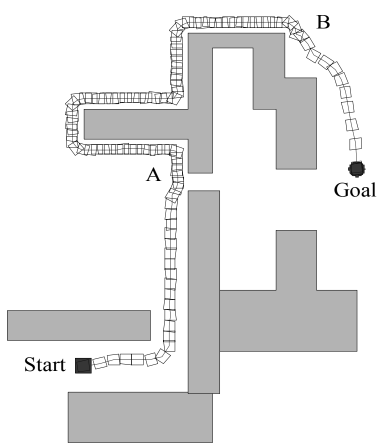
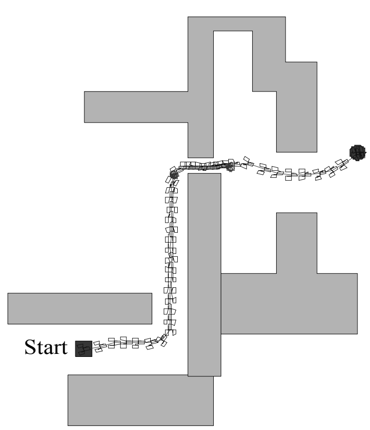


Fig. 8. Robot's passing through the narrow space by reconfiguration

In Fig.8, the proposed method can automatically adjust adaptation factor λ according to the intensity of robot's surrounding obstacles in path planning. When surrounding obstacles of the robot becomes more intensive, the forward distance planned by the algorithm at each time becomes smaller. And candidate positions automatically become intensive to ensure own security of the robot. When there are fewer obstacles around the robot, the forward distance planned by the algorithm at each time becomes larger. So the number of iterations may be reduced and the time of path planning is shortened to improve the planning efficiency. In addition, by using this method, the distance between the robot and obstacles can be maintained larger than the safe distance. The robot can pass through from the middle position when the distance between the two obstacles is small. It can also find the narrower space and change the configuration to pass through, so the adaptability of the method is displayed.



(a)

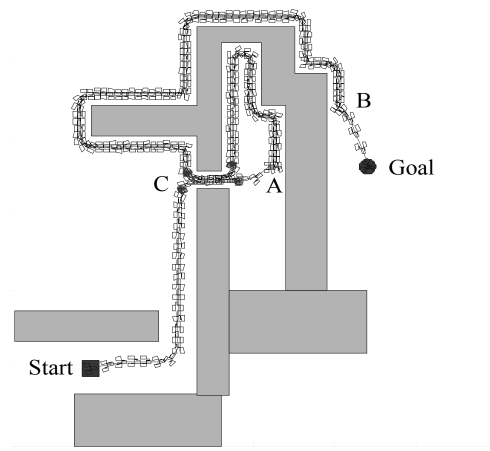


(b)

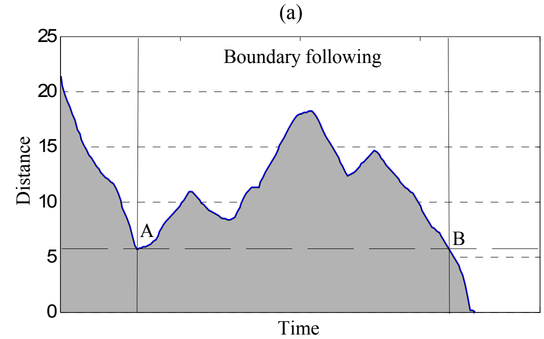
Fig. 9. Comparison of two kinds of robot path planning

In Fig. 9(a), the shape-shifting robot is confined to the configuration "T", thus it degenerates to a fixed-shape robot and can not change its shape to adapt to the environment. In path planning, the deadlock situation occurs at the point A. The robot can successfully circumvent the blocking obstacles by using the *Boundary Following* method. When a point B, which satisfies that d_{BG} is less than D_{block}, is found, the robot finishes the *Boundary Following* of obstacles.

In Fig. 9(b) the shape-shifting robot with the reconfiguration ability can change its own configuration to pass through the narrower space according to the environmental variation, so the *Boundary Following* of obstacles is avoided and the length of path is reduced effectively. Finally, the shape-shifting robot successfully completes the path planning.



(a)



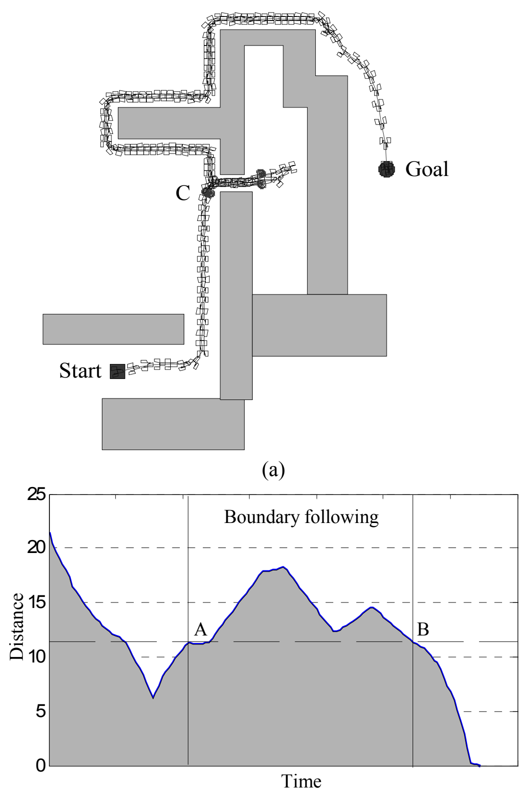
(b) Distance function

Fig. 10. Solving deadlock situation in path planning

The path planning is done by integrating the adaptive reconfiguration of the robot and the solution of the deadlock situation in Fig. 10(a). The robot can not only change configuration according to environmental variation, but also solve the deadlock situation. Fig. 10(b) shows the distance function between the position of the robot and the target position in the whole process of the path planning.

In Fig. 10(a) the robot with the reconfiguration ability can reach the target position if the target position is the point A, while the fixed-shape robot can not reach this position. Therefore, by using the proposed method, the shape-shifting robot can completes the task, which the fixed-shape robot can not complete.

As shown in Fig. 10(a), after the robot changes the configuration to pass through the narrow space at the point C, the deadlock situation occurred at the point A and the distance between the point A and the point C is in a specified range. After the robot chooses the reconfiguration in the reconfiguration position, we use the *Reconfiguration Memory* to resolve such problems if the deadlock situation occurs in a specified range. The robot can quickly escape from the deadlock region by using this method, so unnecessary trouble is reduced and the benefits, which the robot gets shorter path length, are possibly displayed.



(b) Distance function

Fig. 11. The escape of deadlock according to reconfiguration memory

Based on the above analysis, a new path planning method with the *Reconfiguration Memory* is implemented. As shown in Fig. 11(a), the deadlock situation occurs after the robot changes the configuration to pass through the narrow space. The distance between the local deadlock position and the reconfiguration position is less than l (l=5m), so the constraint of the *Reconfiguration Memory* is satisfied. The robot takes the reconfiguration position C as the new target position. When the robot reaches the position C, the reconfiguration evaluation η_1 is set as 1 and the target changes into original target G. Therefore, the robot does not choose reconfiguration here but selects the *Boundary Following*. Finally, the robot reaches the target successfully.

By comparing Fig. 10(a) with Fig. 11(a), when the deadlock situation occurs the robot quickly returns to the position of the *Reconfiguration Memory*, after there is the function of the *Reconfiguration Memory* in the algorithm. So unnecessary *Boundary Following* is reduced.

V. Conclusion

According to the characteristics of the shape-shifting robot, we present an effective path planning method. This method can automatically adjust the moving strategy according to the distribution of obstacles that are around the robot. If there are fewer obstacles around the robot, the method focuses on the rapid movement of the robot. Otherwise, the method focuses on finding a secure mobile position for the robot, when surrounding obstacles become more intensive. By using the *Boundary Following* and evaluating the distance between the robot and the target, the deadlock condition is avoided. In the face of a narrow space, the proposed method can utilize the reconfiguration ability of the robot to ensure the robot pass through the narrow space by changing configuration automatically, and the reconfiguration situation is stored through the *Reconfiguration Memory*. Therefore, this method can reduce the *Boundary Following* and shorten the length of the path, and the unique accessibility of the shape-shifting robot is displayed. Finally, the validation of the proposed path planning method for the shape-shifting robot is demonstrated by simulations.

References

[1] J. Y. Hwang, and J. S. Kim, "A fast path planning by path graph optimization," *IEEE Transactions on Systems, Man and Cybernetics,* Vol.33, No.1, pp. 121-128, 2003.

[2] T. Balch and R. Arkin, "Avoiding the past: a simple but effective strategy for reactive navigation," *IEEE International Conference on Robotics and Automation*, vol. 1, Atlanta, GA, USA, pp. 678-685, May, 1993.

[3] J. A. Janet, R. C. Luo, and M. G. Kay, "The essential visibility graph: an approach to global motion planning for autonomous mobile robots," *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, Vol. 2, pp. 1958-1963, May, 1995.

[4] J. H. Lim and D. W. Cho, "Experimental investigation of mapping and navigation based on certainty grids using sonar sensors," *Robotica*, Vol. 11, No.1, pp.7-17, 1993.

[5] S. Liu, L. Mao and J. Yu, "Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems," *IEEE International Conference on Mechatronics and Automation*, Zhengzhou, China, pp. 1733-1738, June, 2006.

[6] G. Nagib and W. Gharieb, "Path planning for a mobile robot using genetic algorithms," *IEEE International Conference on Electrical, Electronic and Computer Engineering*, Cairo, Egypt, pp.185-189, September, 2004.

[7] Q. Zhang, "A hierarchical global path planning approach for AUV based on genetic algorithm," *IEEE International Conference on Mechatronics and Automation* Zhengzhou, China, pp.1745-1750, June, 2006.

[8] F. Mastrogiovanni, A. Sgorbissa and R. Zaccaria, "Robust navigation in an unknown environment with minimal sensing and representation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 39, No. 1, pp. 212-229, 2009.

[9] J. Jaafar, E. McKenzie and A. Smaill, "A fuzzy action selection method for virtual agent navigation in unknown virtual environments," *IEEE International Conference on Fuzzy Systems* London, U.K., pp.1-6, July, 2007.

[10] J. Velagic, B. Lacevic and N. Osmic, "Efficient path planning algorithm for mobile robot navigation with a local minima problem solving," *IEEE International Conference on Industrial Technology*, Mumbai, India, pp. 2325-2330, December, 2006.

[11]M. A. Wolfensberger, and D. Wright, "Synthesis of reflexive algorithms with intelligence for effective robot path planning in unknown environments," *SPIE Mobile Robots*, Vol.2058, pp.70-81, 1993.

[12]B. Li, S.G. Ma and J.G. Liu. "AMOEBA-I: A Shape-Shifting Modular Robot for Urban Search and Rescue," *Advanced Robotics*, Vol.23, No.9, pp.1057-1083, 2009.

[13]J. Liu, S. Ma, Y. Wang, and B. Li, "Reconfiguration routes research for a self-reconfigurable robot based on network," *Science in China Series F: Information Sciences*, Vol.51, No10, pp.1532-1546, 2008.